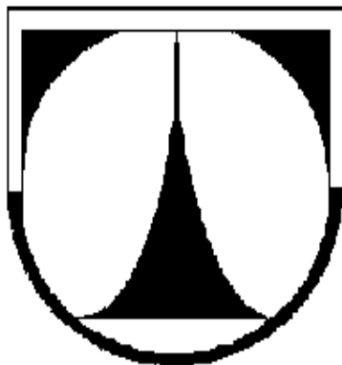


TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií



DIPLOMOVÁ PRÁCE

Návrh wavelet analyzátoru pomocí signálového procesoru

Design of wavelet analyzer using signal processor

Liberec 2004

Radim Kalousek

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Katedra měření



Studijní program M 2612T – Elektrotechnika a informatika

Studijní obor: 3902T005 – Automatické řízení a inženýrská informatika

Návrh wavelet analyzátoru pomocí signálového procesoru

Design of wavelet analyzer using signal processor

Radim Kalousek

Vedoucí diplomové práce:

Doc. Ing. Ivan Jaksch, CSc.

Konzultant:

Ing. Jiří Mareš

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Katedra: měření

Akademický rok: 2003/2004

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro: **Radima Kalouska**

studijní program: M 2612 – Elektrotechnika a informatika

obor: 3902T005 – Automatické řízení a inženýrská informatika

Vedoucí katedry Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto diplomovou práci:

Název tématu:

Návrh wavelet analyzátoru pomocí signálového procesoru

Zásady pro vypracování:

1. Srovnejte aspekty frekvenční a časově-frekvenční analýzy. Vysvětlete nejznámější přístupy jejich výpočtu.
2. Teoreticky zhodnoťte vlastnosti spojitě wavelet transformace.
3. Odvoďte tvar diskrétní wavelet transformace (DWT) užívající číslicové filtrace.
4. Navrhněte a na signálovém procesoru ADSP21061 naprogramujte algoritmus výpočtu DWT.
5. Na modelových signálech ověřte funkčnost takového analyzátoru.

Poděkování

Na tomto místě bych chtěl poděkovat Doc. Ivanu Jakschovi za odborné vedení, pomoc při zpracování diplomové práce a Ing. Jiřímu Marešovi za cenné rady, poskytnuté informace, lidskou podporu a trpělivost.

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č.121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsme vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci

.....

Návrh wavelet analyzátoru pomocí signálového procesoru

Design of wavelet analyzer using signal processor

Resumé:

Diplomová práce popisuje teoreticky a prakticky postup návrhu wavelet analyzátoru. Teoreticky rozebírá vlastnosti a metody frekvenční transformace a časově frekvenční transformace, jejich výhody, uplatnění a vzájemné srovnání. Zvláště je popsána metoda řešící problémy rozlišení v časové, frekvenční a časově frekvenční oblasti – transformace wavelet. Od vlastností spojité wavelet transformace je odvozen tvar diskrétní wavelet transformace, jehož algoritmus využívající číslicové filtrace je naprogramován na signálovém procesoru ADSP21061, vývojové desky EZ-KIT. Funkčnost tohoto analyzátoru byla ověřena na modelových signálech. Grafickou prezentaci výsledků, nastavení a běh analýzy zajišťuje program terminálu realizovaný na PC.

Abstract:

This diploma thesis describes theoretical and practical aspect of design of wavelet analyzer. Theoretically it analyses characteristics and methods of frequency transform and joint time-frequency transform, their advantages, uses and comparison. Especially there is described a method which solves problems of resolution in time, frequency and joint time-frequency domain – the wavelet transform. The form of the discrete wavelet transform is derived from properties of continue wavelet transform and its algorithm using discrete filtering is programmed on signal processor ADSP21061 of evaluation board EZ-KIT. The functionality of this analyzer was verified on model signals. Graphic presentation of results, setting and process of analysis is provided by program of data terminal realized on PC.

Obsah:

Tabulka symbolů	8
Přehled použitých zkratk	9
Seznam obrázků	10
Seznam tabulek	11
Seznam příloh	11
1. Úvod	12
2. Frekvenční analýza	13
2.1. Fourierova transformace	14
2.1.1. Fourierovy řady	14
2.1.2. Fourierova transformace obecného signálu	14
2.1.3. Diskrétní Fourierova transformace	15
2.1.4. Rychlá Fourierova transformace	16
2.1.5. Využití FFT pro frekvenční analýzu periodických signálů	17
2.1.6. CPB analýza	17
3. Časově frekvenční analýza	19
3.1. Krátkodobá Fourierova transformace	21
3.1.1. Volba časového okna	22
3.2. Wiegner-Ville transformace	26
4. Wavelet transformace	29
4.1. Analýza s násobným rozlišením (MRA)	29
4.2. Spojitá waveletová transformace	30
4.2.1. Vlastnosti CWT	31
4.2.2. Výpočet CWT	32
4.3. Diskrétní waveletová transformace	33
4.3.1. Číslicové filtry	33
4.3.2. Banky filtrů	35
4.3.3. Výpočet DWT	37
4.3.4. Wavelet pakety	39
4.4. Wavelet funkce	39
4.5. Některé používané wavelety	40
4.5.1. Časově frekvenční rozlišení wavelet funkcí	42
4.5.2. Výběr waveletu	43
5. Číslicové signálové procesory pro efektivní zpracování signálů	44
5.1. Signálový procesor ADSP21061	44
5.1.1. Základní struktura	45
5.2. Vývojová deska SHARC EZ-KIT Lite	49
5.3. Programování	50
5.3.1. Využití programové prostředí	51
6. Realizace wavelet analyzátoru	52

6.1. Algoritmy DWT.....	52
6.2. Algoritmus DWT pro ADSP21061.....	53
6.2.1. Funkce fir.....	54
6.2.2. Filtry DWT	55
6.3. Možnosti grafického zobrazení.....	56
7. Praktická měření	58
8. Závěr a zhodnocení.....	62
Literatura:	63
Příloha 1:.....	65
Výpis důležitých funkcí programu DWT pro ADSP21061	65

Tabulka symbolů

$x(t), x[n]$	Obecný spojitý, diskrétní vstupní signál
$y(t), y[n]$	Obecný spojitý, diskrétní výstupní signál
$F[k]$	Komplexní koeficient Fourierovy řady
T	Perioda signálu
j	Imaginární jednotka
n, k	Pořadnice diskrétní posloupnosti v čase, ve frekvenci
φ	Fázový úhel
P_x	Výkon signálu
$F\{\}, F^{-1}\{\}$	Fourierova, inverzní Fourierova transformace
A, ω	Amplituda, úhlová frekvence harmonického signálu
$X(j\omega)$	Obraz Fourierovy transformace
N	Počet vzorků
Δf	Frekvenční krok
W_n	Otáčecí činitel FFT
$X[k]$	Komplexní koeficient DFT
f_c, f_d, f_h	Centrální, dolní, horní frekvence pásmové propusti
$\Delta t, \Delta \omega$	Časový, frekvenční interval
B	Šířka frekvenčního pásma filtru
τ	Časové posunutí
$w(t)$	Okénková funkce
s	Koeficient CWT vyjadřující měřítko
ψ	Mateřský wavelet
z^{-1}	Zpožďovací člen
$\Psi_x^\psi, \Psi_{k,p}$	Koeficienty CWT, DWT
$\hat{x}[n]$	Diskrétní signál na výstupu banky filtrů
DP, HP	Dolní propust, horní propust
H, G	Přenosové funkce filtrů dolní, horní propusti
$h[n], g[n]$	Impulsní odezva filtru dolní propusti, horní propusti
\bar{h}, \bar{g}	Impulsní odezva filtru dolní propusti, horní propusti rekonstrukčních filtrů IDWT
M	Počet frekvenčních pásem
c	Kalibrační konstanta

tab.1 Tabulka symbolů

Přehled použitých zkratek

ALU (Arithmetic Logic Unit) – aritmeticko logická jednotka.

ANSI (American National Standards Institute) – americká instituce, která vyvíjí americké průmyslové standardy ve shodě s mezinárodními standardy ISO.

CPB (Constant Percentage Bandwidth) – analýza s konstantním procentuálním krokem pásma.

CPU (Central Processing Unit) – centrální procesorová jednotka provádí výpočty a řídí překlad i vykonávání instrukcí.

CWT (Continuous Wavelet Transform) – spojitá waveletová transformace.

DAG (Data Address Generator) – generátor datových adres.

DFT (Discrete Fourier Transform) – diskrétní Fourierova transformace.

DMA (Direct Memory Access) – metoda přenosu dat mezi operační pamětí a dalším zařízením bez účasti CPU.

DP – dolní propust.

DSP (Digital Signal Processor) – digitální signálový procesor.

DWT (Discrete Wavelet Transform) – diskrétní waveletová transformace.

EPROM (Erasable Programmable Read Only Memory) – přepisovatelná paměť ROM, jejíž obsah se neztrácí při výpadku elektrického proudu, ale lze jej měnit i po výrobě, ne však prostředky počítače.

FFT (Fast Fourier Transform) – rychlá Fourierova transformace.

FIR (Finite Impulse Response) – číslicový filtr s konečnou impulsní odezvou.

FLAG – příznak či identifikátor. Obecně je to paměťová buňka určená pro uložení informace o stavu nebo existenci nějakého jevu nebo procesu v systému.

FT – Fourierova transformace.

HP – horní propust.

I/O (Input/Output) – vstupně-výstupní.

IDFT (Inverse Discrete Fourier Transform) – inverzní diskrétní Fourierova transformace.

IDWT (Inverse Discrete Wavelet Transform) – inverzní diskrétní waveletová transformace.

IEEE (Institute of Electrical and Electronics Engineers) – americká instituce definující normy a standardy, které se týkají především elektrických parametrů zařízení.

- IIR** (Infinite Impulse Response) – číslicový filtr s nekonečnou impulsní odezvou.
- LED** (Light Emitting Diode) – polovodičové zařízení produkující světlo.
- MRA** (Multiresolution Analysis) – analýza s násobným rozlišením.
- MW** (Mother Wavelet) – mateřský wavelet.
- PROM** (Programmable Read Only Memory) – druh paměti ROM, která je pouze jednorázově programovatelná.
- RAM** (Random Access Memory) – paměť s přímým přístupem, umožňuje zápis i čtení.
- RS232** (Recommended Standard 232) – americká forma definující konektory, linky, propojení a vlastnosti signálu při sériovém komunikačním přenosu.
- SHARC** (Super Harvard Architecture) – druh architektury uspořádání paměti.
- SRAM** (Static RAM) – statická paměť RAM, je založena na stavech klopných obvodů, které vyžadují k uchování informace velmi málo elektrického proudu ve srovnání s dynamickou RAM.
- STFT** (Short Time Fourier Transform) – krátkodobá Fourierova transformace.
- UART** (Universal Asynchronous Receiver Transmitter) – architektura elektronického obvodu určeného pro vysílání a příjem dat.
- WT** – waveletová transformace.
- WVT** – Wiegner Ville transformace.

Seznam obrázků

obr.2.1 Porovnání rychlosti DFT a FFT	17
obr.2.2 Specifikace třetino-oktávové CPB	18
obr.3.1 Rozdělení časově frekvenčních postupů	19
obr.3.2 Příklad časově frekvenční analýzy	20
obr.3.3 Časový průběh obdélníkového okna	22
obr.3.4 Spektrum obdélníkového okna	23
obr.3.5 Signál vážený obdélníkovým oknem.....	23
obr.3.6 Časový průběh Hanningova okna.....	24
obr.3.7 Spektrum Hanningova okna	24
obr.3.8 Časový průběh Flat Top okna.....	25
obr.3.9 Spektrum Flat Top okna	25
obr.3.10 Signál vážený Gaussovým oknem.....	26
obr.4.1 Časově-kmitočtové rozlišení WT a STFT	29
obr.4.2 Přímá struktura FIR filtru – transversální filtr	34
obr.4.3 Stromová struktura vytvářející lineární banku filtrů	35
obr.4.4 Kaskádní struktura vytvářející exponenciální banku filtrů	35
obr.4.5 Základní blokové schéma dvoukanálové banky filtrů.....	36
obr.4.6 Frekvenční pohled na diskrétní wavelet transformaci.....	37
obr.4.7 Mexican hat	40
obr.4.8 Morletův wavelet.....	40

obr.4.9 Mayerův wavelet	41
obr.4.10 Konvoluční filtry pro výpočet DWT pomocí Mayerova waveletu	41
obr.4.11 Haarův wavelet	42
obr.4.12 Wavelet Daubichies	42
obr.5.1 Blokové schéma vnitřní struktury procesoru ADSP21061	45
obr.5.2 Vývojová deska SHARC EZ-KIT Lite	49
obr.5.3 Blokové schéma SHARC EZ-KIT Lite	50
obr.6.1 Snižování počtu vzorků při DWT	52
obr.6.2 Pořadí při zpracování dat DWT	53
obr.6.3 Měřitková funkce a wavelet Daubechies 8, a koeficienty filtrů	56
obr.6.4 Zobrazení wavelet analýzy pomocí scalogramu	57
obr.6.5 Zobrazení aproximační a detailních složek signálu (3 stupně dekompozice)	57
obr.7.1 Necejchovaná wavelet analýza harmonického signálu 6kHz	59
obr.7.2 DWT lineárně frekvenčně přelad'ovaného sinusového signálu	59
obr.7.3 DWT logaritmicky frekvenčně přelad'ovaného sinusového signálu	60
obr.7.4 Časový průběh DWT koeficientů z první úrovně dekompozice	61

Seznam tabulek

tab.1 Tabulka symbolů	8
tab.3.1 Porovnání vlastností časových oken	26
tab.4.1 Časové, frekvenční a časově frekvenční rozlišení vybraných wavelet funkcí	42
tab.p1 Tabulka řídicích registrů SPORTx	65
tab.p2 Tabulka registrů AD1847	69

Seznam příloh

Příloha 1: Výpis důležitých funkcí programu DWT pro ADSP21061	65
---	----

Přílohy obsažené na disku CD-R:

CD-R obsahuje následující strukturu adresářů:

Dp – vlastní text diplomové práce.

Dwt – komentovaný zdrojový kód programu wavelet analyzátoru pro ADSP21061

DWT.C s potřebnými hlavičkovými soubory: *COEFFD8_HP.H*, *CO-EFFD8_LP.H*, *UROVNE.H*.

Stupne_konvoluce – soubor *STUPNE.EXE* generující hlavičkový soubor *UROVNE.H* a jeho zdrojový text v jazyce C.

1. Úvod

Při zpracování signálu používáme určité transformace z toho důvodu, abychom získali nový pohled na signál a mohli zřetelněji vystihnout jeho vlastnosti, které nám mají pomoci k jeho efektivnějšímu zpracování. Pojem efektivnost může být myšlen kratší čas zpracování, menší paměť nutná pro uložení signálu, odstranění šumu, zvýraznění či potlačení specifických rysů signálu apod. Klasickou a velmi používanou transformací je Fourierova transformace, která má svoji diskrétní podobu v diskrétní Fourierově transformaci (Discrete Fourier Transform – DFT).

Vznik wavelet transformace (WT) je spojen se jménem francouzského geofyzika Jeana Morleta, který ji vymyslel na začátku 80. let jako nástroj pro analýzu seismických signálů. Tato transformace se těšila již od svého počátku velké přízni a číslorodému zájmu jak matematiků, tak i zájemců o její využití při zpracování signálů a obrazové informace v nejrůznějších oborech. Podnětem pro vznik WT byla snaha získat časově-frekvenční popis signálu. Fourierova transformace poskytuje informaci o frekvenčních složkách signálu, ale nevypovídá o jejich poloze v čase. Wavelet transformace získání takového popisu umožňuje.

Není ovšem cílem touto transformací nahradit používání klasických transformací, jako je například již zmíněná Fourierova transformace, ale díky WT jsou rozšířeny možnosti výběru transformace podle typu aplikace. Každá z analýz je vhodná pro jiný druh diagnostického signálu. Úkolem této diplomové práce je také teoretické srovnání přístupů frekvenční a časově-frekvenční analýzy.

Realizace algoritmu diskrétní wavelet transformace bude stavět na platformě signálového procesoru ADSP21061, vývojové desky EZ-KIT [1,2,3].

2. Frekvenční analýza

Průběh signálu se obvykle znázorňuje v čase, jinak řečeno v časové oblasti – doméně. Posuzování časového průběhu signálu vhodně doplňují i jeho vlastnosti prezentované rozkladem na soubor elementárních funkcí. Nejpřirozenější je rozklad na soubor harmonických funkcí, které se liší amplitudou, úhlovou frekvencí a svou počáteční fází. Mimo to je v praxi často výhodné (teoreticky i experimentálně) používat harmonických funkcí $\exp(i\omega t)$, neboť jsou snadno prakticky realizovatelné (resp. jejich imaginární a reálná část) a mají výhodné matematické vlastnosti zvláště vzhledem k derivaci a integrování.

Jestliže se u souboru harmonických signálů znázorní závislost amplitudy a počáteční fáze na frekvenci, pak je signál znázorněn ve frekvenční oblasti – doméně. Rozklad periodické funkce se spojitým časem na kombinaci harmonických signálů se nazývá Fourierova (nekonečná) řada. Pro obecné neperiodické funkce se používá Fourierova transformace (FT).

Frekvenční oblast je zvláště vhodná pro analýzu periodických nebo kvasiperiodických signálů. Složení tohoto typu signálu se analyzuje ve frekvenční oblasti mnohem přehledněji než v časové oblasti.

Definiční vzorec pro FT je integrálem a pro praktickou realizaci není příliš vhodný, neboť jeho analytické řešení existuje jen v omezeném počtu případů a je nutno jej tedy řešit přechodem z nekonečného integrálu na konečnou sumaci. V případě počítačového zpracování nemáme spojitou funkci, ale jen její hodnoty v diskrétních vzorkovacích okamžicích. Z těchto důvodů se definuje diskrétní Fourierova transformace (DFT), která je již polynomem a jejími vstupy a výstupy jsou posloupnosti hodnot. Nevýhodou této definice je její značná časová náročnost, která roste se čtvercem délky její vstupní posloupnosti. Proto byl vypracován algoritmus, který vychází z vlastností exponenciálních diskrétních funkcí a výrazně snižuje potřebnou dobu výpočtu. Tento algoritmus je zvykem nazývat rychlá Fourierova transformace (FFT).

Fourierova transformace se ukázala být účinnou metodou zpracování různých signálů. Často je využíváno její vlastnosti převodu konvoluce na násobení, což umožňuje zavést tzv. přenosovou frekvenční funkci, která vhodným způsobem charakterizuje dynamické vlastnosti soustavy. Metoda umožňuje provádět frekvenční filtraci, tedy odstraňovat ze signálu části s různými frekvencemi, což může např. snížit úroveň šumu v signálu. Operace ve frekvenční oblasti mohou například i upravovat obrazy takovým

způsobem, aby došlo k zvýraznění hran, k odstranění „proužkování“, ke zvýraznění některých struktur... [4,5].

2.1. Fourierova transformace

2.1.1. Fourierovy řady

Nejjednodušší odvození Fourierovy transformace vychází z tzv. Fourierovy řady periodické funkce. Periodická funkce je charakterizována rovností vzájemně posunutých funkčních hodnot $x(t) = x(t + nT)$, kde T je perioda a $n = \pm 1, \pm 2, \pm 3, \dots$ je její násobek. Definiční vzorce Fourierovy nekonečné řady jsou následující:

$$F[k] = \frac{1}{T} \int_0^T x(t) e^{-j \frac{2\pi}{T} kt} dt, \quad (2.1)$$

$$x(t) = \sum_{k=-\infty}^{+\infty} F[k] e^{j \frac{2\pi}{T} kt}, \quad (2.2)$$

kde $F[k]$, $k = 0, \pm 1, \pm 2, \dots$ jsou koeficienty Fourierovy řady.

Fourierova řada představuje rozklad signálu na nekonečný počet dvojic vektorů, které rotují proti sobě. Umožňuje rozložit libovolný periodický signál na harmonické složky, které jsou tvořeny harmonicky vázanými (ko)sinusoidami, jejichž frekvence je celistvým násobkem opakovací frekvence signálu. Z harmonických složek je možné zpětně signál zrekonstruovat podle vztahu 2.2.

Mezi komplexní funkcí $x(t)$ a koeficienty $F[k]$ platí Parsevalův vztah:

$$P_x = \frac{1}{T} \int_0^T |x(t)|^2 dt = \sum_{k=-\infty}^{+\infty} |F[k]|^2, \quad (2.3)$$

který udává výkon signálu jak v časové tak i ve frekvenční oblasti a na jeho základě lze určit, kolik harmonických složek je třeba k dostatečnému popisu signálu (čím více složek, tím lepší zpětná rekonstrukce signálu) [4,6].

2.1.2. Fourierova transformace obecného signálu

Rozklad na Fourierovu řadu se týká jen periodických signálů. Přestože počet složek rozkladu je obecně nekonečný, obsahuje tento rozklad jen složky s frekvencemi, které jsou násobky, tzv. harmonické, základní frekvence opakování signálu. To znamená, že spektrum obsahuje jen izolované složky.

Rozklad obecného, tj. nejen periodického signálu, ale také neperiodického signálu na harmonické složky, lze vypočítat s pomocí Fourierovy transformace. Výraz pro Fourierovu transformaci můžeme odvodit z Fourierovy řady rozšířením intervalu periodicity T na $(-\infty, \infty)$. Tento rozklad obsahuje obecně složky o všech frekvencích. Spektrum je spojitá funkce frekvence. Definiční vzorce přímé a zpětné (inverzní) Fourierovy transformace pro signál, tj. funkci $x(t)$ ve významu vzoru nebo originálu jsou následující:

$$X(j\omega) = F\{x(t)\} = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt, \quad (2.4)$$

$$x(t) = F^{-1}\{X(j\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(j\omega)e^{j\omega t} d\omega, \quad (2.5)$$

kde funkce úhlové frekvence $X(j\omega)$ má význam obrazu nebo také obecně signálu, který je transformací do frekvenční oblasti a je nazýván Fourierovo nebo komplexním spektrem.

Fourierova transformace je zobrazení na prostoru komplexních funkcí, a proto je Fourierův obraz reálné funkce obecně funkce komplexní. Lze ho rozložit na amplitudu a fázi

$$F(x(t)) = |F| e^{j\varphi}, \quad (2.6)$$

kde amplituda $|F|$ je Fourierovo spektrum vyšetřované funkce a φ je fázový úhel. Spektrum obsahuje informace o frekvencích přítomných v signálu a fázový úhel nese informaci o vzájemném posunutí rozkladových funkcí vůči počátku. Signál je tedy charakterizován nelokálními funkcemi a jejich vzájemným posunutím. Pro aplikace zaměřené na lokální vlastnosti signálu není tento popis zcela vhodný. Vhodnější se jeví popis pomocí lokalizovaných funkcí, tj. funkcí jimž lze přisoudit polohu v čase. Rozklad signálu je v těchto případech místo frekvence a fáze charakterizován frekvencí a polohou v čase a zabývá se jím tzv. časově frekvenční analýza [4,7].

2.1.3. Diskrétní Fourierova transformace

Uvedené tvary stále popisují pouze matematickou podstatu. Při zpracování signálů pomocí počítačů a speciálních číslicových integrovaných obvodů se využívá konečný počet vzorků a u spojitých funkcí lze pracovat pouze se vzorky těchto funkcí. Přitom signály v oblasti času i frekvence mají konečný počet hodnot N a při výpočtech se

považují za periodické (pracuje se s periodickým prodloužením průběhu ze základního intervalu). Transformace umožňující přechody mezi časovou oblastí, kde nezávisle proměnnou budeme značit n , a frekvenční oblastí, kde nezávisle proměnnou budeme značit k , se nazývá diskrétní Fourierova transformace (DFT) a je definována vztahy:

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} nk}, \quad (2.7)$$

$$x[n] = \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} nk} \quad (2.8)$$

Vztah (2.7) se nazývá přímá diskrétní Fourierova transformace (DFT) a vztah (2.8) zpětná (inverzní) diskrétní Fourierova transformace (IDFT).

Výsledkem DFT je konečný počet diskrétních vzorků. Signálu $x[n]$ o N vzorcích, tj. posloupnosti $x[0], x[1], \dots, x[N-1]$ odpovídá N frekvenčních vzorků $X[k]$ [8,9].

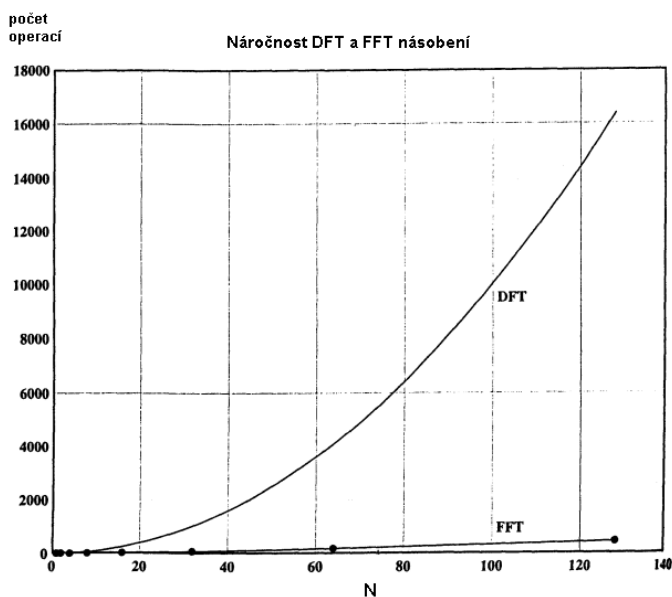
2.1.4. Rychlá Fourierova transformace

Do 60. let bylo využití DFT dle (2.7) a (2.8) omezeno na sálové počítače a výpočty pro velké objemy dat trvaly desítky minut i hodin. Pro výpočet N hodnot $X[k]$ podle (2.7) je totiž třeba N^2 komplexních násobení a $N(N-1)$ (obecně komplexních) sčítání, čili doba potřebná pro výpočet je přibližně úměrná N^2 .

Rychlá Fourierova transformace (Fast Fourier Transform – FFT) je velmi efektivní způsob výpočtu DFT. Byl popsán v roce 1965 J.W.Cooleyem a J.W.Tukeym a znamenal revoluci v číslicovém zpracování signálu. FFT je definována vztahem:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk} \quad (2.9)$$

Obdobné algoritmy byly objeveny již na přelomu století, ale nebyly prakticky využity vzhledem k tehdejšímu stavu výpočetní techniky. Dnes existuje těchto algoritmů celá řada. Využívají periodičnosti a symetrií exponenciály v (2.7) a (2.8), kterou bývá zvykem označovat jako otáčecí činitel (twiddle factor) $W_N^{nk} = e^{-j \frac{2\pi}{N} nk}$ a tím vznikají úspory ve výpočtech. Časová náročnost pak odpovídá $N \cdot \log N$ [6,8].



obr.2.1 Porovnání rychlosti DFT a FFT

2.1.5. Využití FFT pro frekvenční analýzu periodických signálů

V důsledku velké výpočtové rychlosti FFT se DFT stala velmi důležitým nástrojem také v měřicí technice, zejména pro frekvenční analýzu a číslicovou filtraci. Výhodou frekvenční analýzy prováděné pomocí FFT proti analýze s užitím číslicových filtrů je, že pomocí FFT získáme nejen amplitudové spektrum (absolutní hodnotu spektra), ale i fázové spektrum. Pomocí těchto spekter je možno výhodně počítat řadu dalších charakteristik signálů (autokorelační funkce, výkonové spektrální hustoty apod.) a u dvoukanálových měření lze pomocí FFT zjišťovat i vzájemné korelační funkce a spektra [8].

2.1.6. CPB analýza

FFT analýza poskytuje spektrum v lineární frekvenční stupnici. Existují však aplikace, kterým vyhovuje stupnice logaritmická (s exponenciálním krokem). Taková osa lépe popíše spektra, které jsou přirozeně logaritmická nebo taková, pro které potřebujeme větší rozlišení na začátku frekvenční osy. Často se proto hodí nástroj představovaný CPB analýzou.

CPB je zkratka anglického constant percentage bandwidth, což v překladu znamená analýzu s konstantním procentuálním krokem pásma ($\Delta f/f_c = \text{konst.}$). V praxi to znamená, že šířka pásma je násobkem diskrétní střední frekvence. Střed propustného

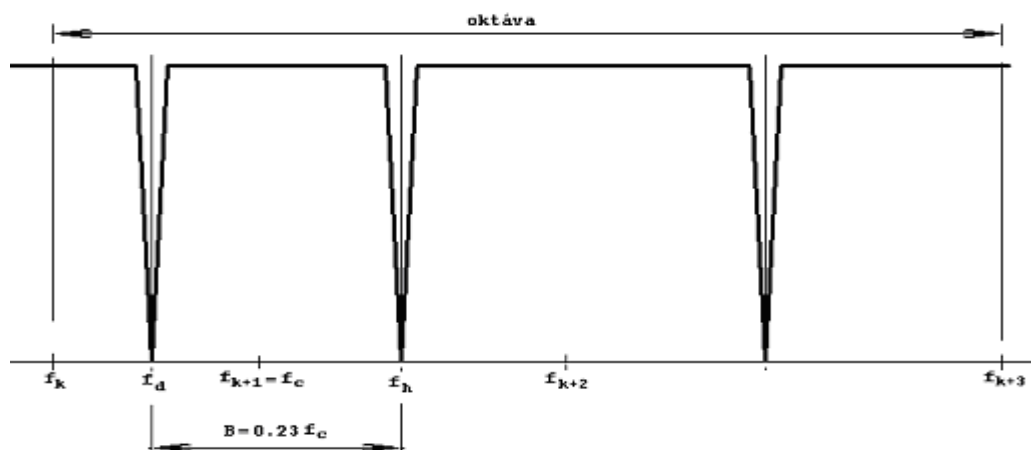
pásma u CPB analyzátorů je definován geometrickým průměrem horní a dolní mezní frekvence $f_c = \sqrt{f_d f_h}$.

Vzdálenost středních frekvencí sousedních pásem je volena jedna oktáva (zdvojnásobení kmitočtu) nebo zlomek oktávy. Rozlišení CPB analýzy je definováno počtem filtrů v oktávě. Nejčastěji se tak můžeme setkat s oktávovou, 1/3-no, 1/6-no, 1/12-no, 1/24-no oktávovou analýzou. Jednotlivé stupně, střední frekvence nebo šířky filtrů, tak vytvářejí geometrickou řadu s kvocientem $2^{1/n}$, kde n je počet pásem v oktávě. Kvůli typizaci byla jako nulová referenční hladina zvolena frekvence 1000Hz. Ostatní střední frekvence tak můžeme dopočítat. U třetino-oktávového dělení tak dostáváme tyto základní vztahy:

$$f_{k+3} = 2f_k, \quad f_{k+1} = 2^{1/3} f_k \quad (2.10)$$

$$B = f_h - f_d = (2^{1/6} - 2^{-1/6})f_k = 0.23 f_k \quad (2.11)$$

Rovnice (2.10) dává jasný předpis pro výpočet libovolné centrální frekvence.



obr.2.2 Specifikace třetino-oktávové CPB

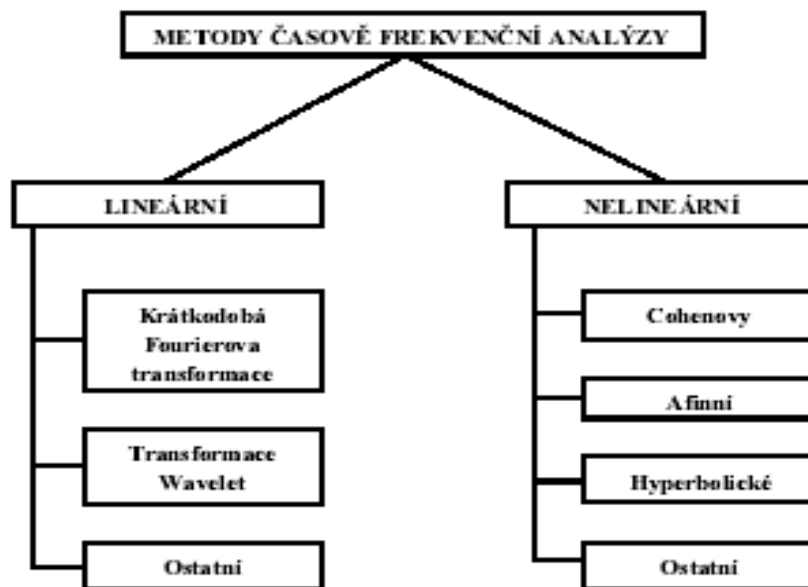
CPB analýza není transformací, neexistuje pro ni teoretický podklad přímou matematickou definicí. Popis se proto opírá o syntézu ze známých metod. Používané jsou dvě cesty. První staví na idealizovaném modelu s exponenciální bankou pásmových propustí, což je model prosazovaný v analyzátorech Brüel&Kjaer. Druhá staví na syntéze z kaskádní FFT a je užívaná např. firmou National Instrument.

Použití CPB analýzy se jeví výhodné pro taková měření spekter, která se vyznačují zásadními projevy na počátku stupnice, která jsou spojitá a širokopásmová. Charakteristická jsou pak nasazení při identifikaci zvukových a mechanických kmitů a měření intenzit [4,10].

3. Časově frekvenční analýza

Je vhodné podotknout, že převážná většina reálných technických signálů je zařazována do oblasti nestacionárních signálů. Fourierova transformace, případně její modifikace jsou techniky zvláště vhodné ke zpracování stacionárních signálů. Mohou být využity i pro analýzu přechodových a nestacionárních signálů, pokud nás zajímají pouze frekvenční komponenty obsažené v celém signálu. Nedávají nám přehled o časovém výskytu frekvenčních složek.

Lze tedy konstatovat, že pro určení časové lokalizace frekvenčních komponent nelze použít klasický postup frekvenční analýzy, ale je nutné využít jiné transformační postupy a jiné výpočetní metody. Jedním z možných postupů, jak analyzovat časový výskyt frekvenčních složek přechodových a nestacionárních signálů, je použití tzv. časově frekvenčních postupů (transformací). Ty mohou být rozděleny podle výpočetního postupu do dvou základních tříd (obr. 3.1).



obr.3.1 Rozdělení časově frekvenčních postupů

Výhodou lineárních transformací je zejména rychlost výpočtu a uspokojivá časově frekvenční rozlišení. Hlavní nevýhodou lineárních transformací je skutečnost, že výsledná rozlišení v čase a frekvenci je limitováno tzv. Heisenbergovým principem neurčitosti (3.1), který říká, že nelze přesně vědět, jaká frekvence se vyskytuje v daném časovém okamžiku.

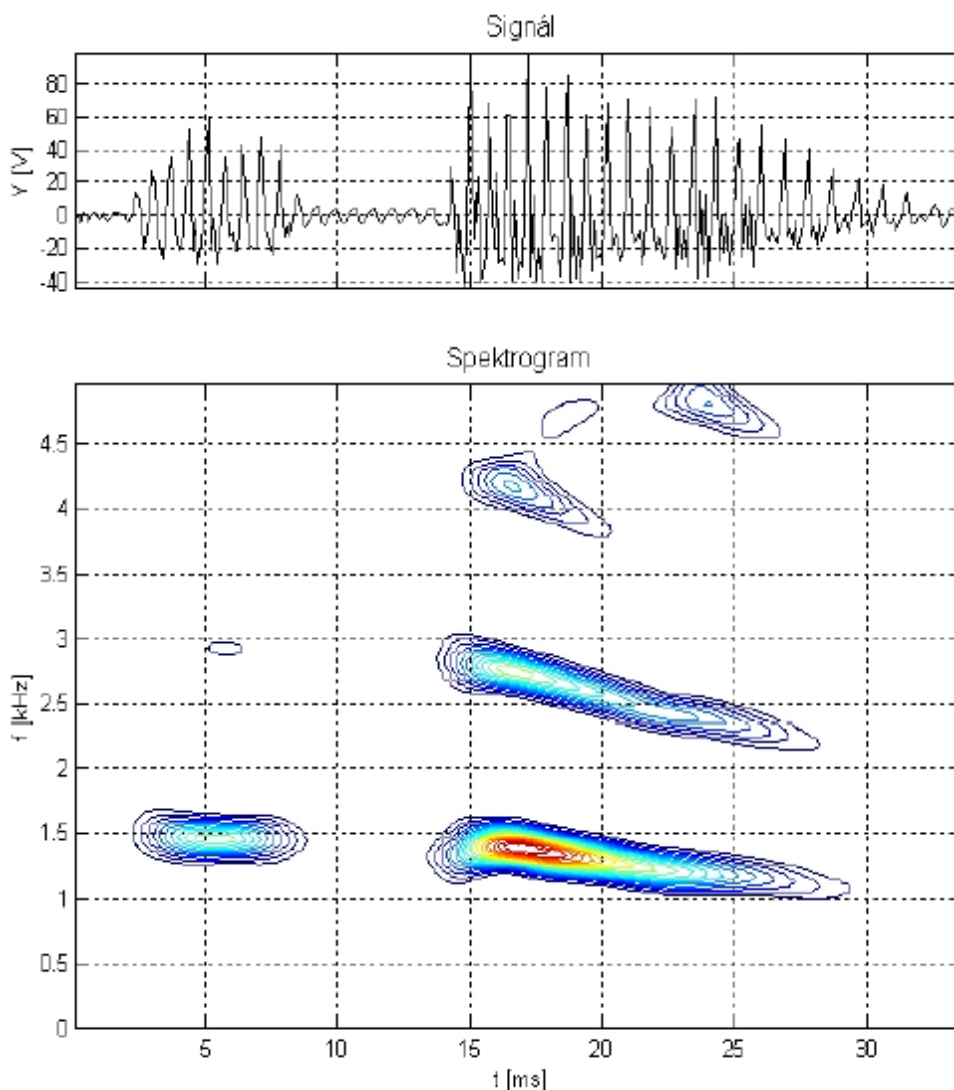
$$\Delta^2 t \cdot \Delta^2 \omega \geq \frac{1}{2} \quad (3.1)$$

Kde platí

$$\Delta^2 t = \int t^2 |x(t)|^2 dt \quad (3.2)$$

$$\Delta^2 \omega = \frac{1}{2\pi} \int \omega^2 |X(\omega)|^2 d\omega \quad (3.3)$$

Důsledkem tohoto principu neurčitosti je, že složka signálu nereprezentuje přímo bod v časově frekvenčním prostoru. Je tedy možné pouze určit pozici obdélníku $\Delta t \cdot \Delta \omega$ v dané časově frekvenční oblasti (Δt představuje minimální časový interval – časový krok, $\Delta \omega$ představuje minimální frekvenční interval – frekvenční krok).



obr.3.2 Příklad časově frekvenční analýzy

Druhou základní třídu časově frekvenčních transformací představují kvadratické metody. Důležitá skupina kvadratických transformací jsou tzv. Cohenovy transformace. Jde o všechny časově frekvenční postupy, které jsou invariantní k času a frekvenci.

Kvadratické časově frekvenční transformace invariantní k času a měřítku tvoří další základní třídu, tzv. afinní transformace.

Charakteristickým rysem všech kvadratických transformací je skutečnost, že jejich výsledné rozlišení v čase a frekvenci není limitováno Heisenbergovým principem neurčitosti. To zajišťuje vysokou rozlišovací schopnost v časově-frekvenční rovině, která se projevuje „přesnou“ lokalizací význačných frekvenčních komponent v čase. Jistou nevýhodou všech nelineárních postupů, zejména při zpracování reálných signálů s velkým počtem vzorků, je časová náročnost výpočtu a nároky na relativně velkou operační i diskovou paměť počítače. Další nevýhodou některých nelineárních transformací při zpracování signálů může být existence „falešných“ interferenčních frekvenčních komponent [11,12].

3.1. Krátkodobá Fourierova transformace

K nejdříve používaným časově frekvenčním postupům patří využití modifikace Fourierovy transformace, nazývané dle postupu výpočtu krátkodobá Fourierova transformace (Short Time Fourier Transform – STFT). STFT lokalizuje frekvenční složky v čase s konstantním rozlišením. Základním principem metody je rozdělení signálu na malé realizace, u nichž je možno předpokládat dostatečnou stacionaritu. To je provedeno multiplikací jisté okénkové funkce a signálu. Na každém takovém výřezu je provedena Fourierova transformace. Okénko se postupně posouvá v čase. STFT poskytuje kompromis mezi časovou a frekvenční reprezentací signálu. Její definiční integrál je dán rovnicí

$$STFT(\tau, f) = \int_{-\infty}^{\infty} [x(t) \cdot w^*(t - \tau)] \cdot e^{-j2\pi \cdot f \cdot t} \cdot dt, \quad (3.4)$$

kde w je okénková funkce, $*$ je komplexní konjugace, τ je časové posunutí okénka, $x(t)$ je časová reprezentace signálu a $STFT(\tau, f)$ je jeho časově frekvenční reprezentace. Rekonstrukci signálu $x(t)$ je možné realizovat zpětnou (inversní) transformací dle vztahu:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} STFT(\tau, f) \cdot w(t - \tau) \cdot e^{j2\pi \cdot f \cdot t} \cdot d\tau \cdot df \quad (3.5)$$

V technické praxi obvykle signál obsahuje význačné frekvenční složky různých řádů. Proto je někdy nevýhodou STFT skutečnost, že se aplikuje časové okno stejné šířky pro všechna frekvenční pásma a tudíž frekvenční oblast je rozdělena lineárně.

Přes určité omezení vyplývající z Heisenbergova principu neurčitosti a z něj pramenících omezení s výběrem vážící okénkové funkce a její šíře, se STFT stává jedním ze základních a rychlých přístupů pro časově frekvenční analýzu stacionárních i nestacionárních signálů.

Přesnost a vhodnost této metody závisí na volbě okénkové funkce, její velikosti a na případném překrytí jednotlivých segmentů. Překrytí zajišťuje, že vzorky na okraji budou zpracovány stejně efektivně jako vzorky uprostřed časového okna [11].

3.1.1. Volba časového okna

Při výpočtu DFT je předpokládáno, že signály jsou periodické. To znamená, že jen pro frekvence harmonického signálu, které jsou násobkem $1/T$, obsahuje záznam celočíselný počet period. Harmonické signály s neceločíselným násobkem své frekvence vzhledem k $1/T$ jsou zaznamenány jako výsek, o kterém je implicitně při výpočtu DFT předpokládáno, že je jednou celistvou periodou signálu.

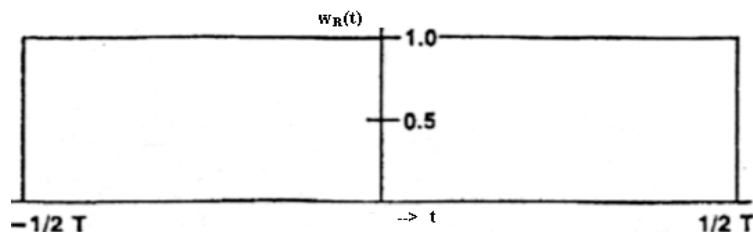
Jelikož se při STFT postupuje tak, že se vstupní signál rozdělí na kratší úseky a z nich jsou následně spočítána místní spektra pomocí Fourierovy transformace, je zřejmé, že tyto úseky nemusí zahrnout přesně celočíselný násobek periody signálu. To se také projeví ve složení spektra, kde vzniknou zdánlivé složky, které ve skutečnosti v harmonickém signálu nejsou.

Rozdělení signálu na menší realizace je provedeno jako součin původního signálu s určitým typem časového okna.

- Obdélníkové (Rectangular) časové okno $w_R(t)$ lze vyjádřit vztahy:

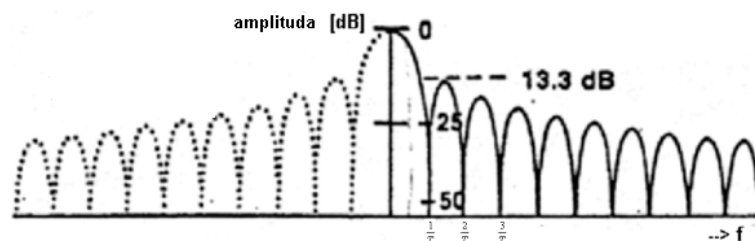
$$w_R(t) = 1 \quad \text{pro} \quad -\frac{T}{2} \leq t \leq \frac{T}{2},$$

$$w_R(t) = 0 \quad \text{pro ostatní } t \quad (3.6)$$



obr.3.3 Časový průběh obdélníkového okna

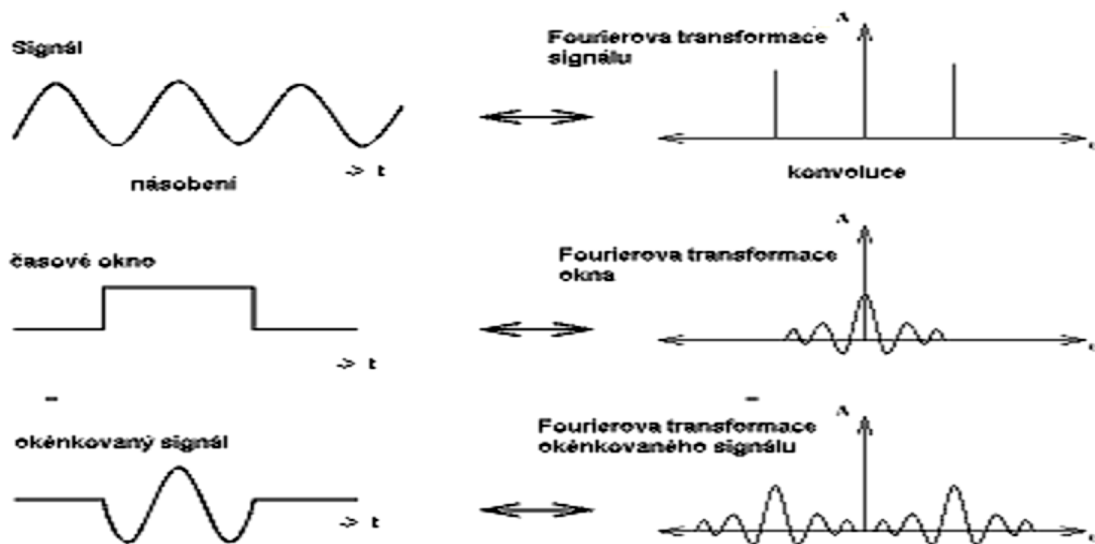
Spektrum obdélníkového okénka je funkce $\text{sinc}(x)$.



obr.3.4 Spektrum obdélníkového okna

Nejvyššímu oblouku se říká "hlavní lalok", ostatní oblouky jsou tzv. "postranní laloky" nebo "postranní vlny". Pro frekvenční analýzu je výhodné, aby postranní laloky byly proti hlavnímu co nejnižší (tím se potlačí rušivé složky spektra) a přitom aby byl hlavní lalok co nejužší (pak se ve spektru objeví minimum velkých rušivých složek blízko analyzované frekvence). Dalším požadavkem je minimalizace největší možné chyby hlavní čáry spektra, čili velikost poklesu amplitudy hlavního laloku. Z hlediska interpretace výsledků měření je také důležitým údajem šířka pásma šumu vyjádřená jako násobek frekvenční vzdálenosti složek spektra. Udává, kolikrát se signál po aplikování časového okna rozšíří ve frekvenční oblasti.

V případě obdélníkového okna mají postranní laloky malý odstup od hlavního laloku. Neleží-li spektrální frekvence na diskrétní frekvenci f_k , $k=0,1,\dots,N/2-1$, je spektrum značně roztaženo a navíc amplituda spektra pro hlavní frekvenci je značně zkrácena.



obr.3.5 Signál vážený obdélníkovým oknem

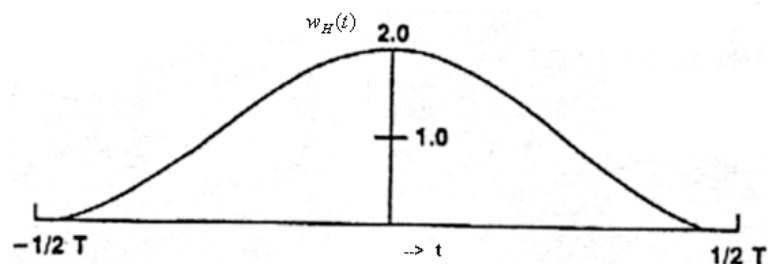
Proto bylo vymyšleno mnoho jiných okének. Jejich spektrum má širší hlavní složku přes dvě a více diskrétních frekvencí a větší odstup postranních laloků od hlav-

ního. To má výhodu v menší chybě amplitudy, na druhé straně však dochází k roztažení hlavního pásma a tak není možné přesně lokalizovat skutečnou frekvenci.

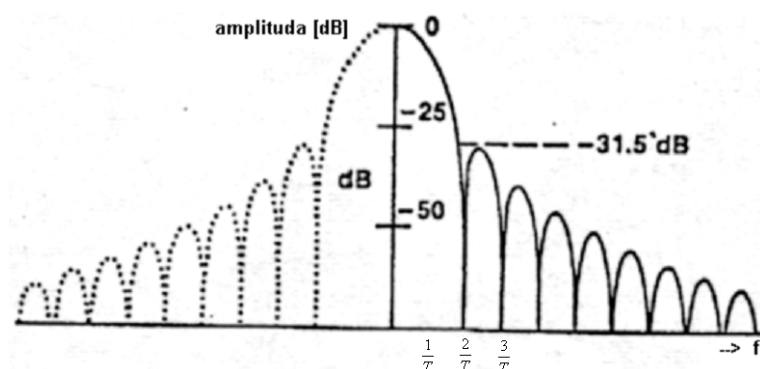
- Okno Hanning je definováno vzorcem

$$w_H(t) = 1 - \cos\left(\frac{2 \cdot \pi \cdot t}{T}\right) \text{ pro } 0 \leq t < T$$

$$w_H(t) = 0 \text{ pro } t < 0, T \leq t \quad (3.7)$$



obr.3.6 Časový průběh Hanningova okna

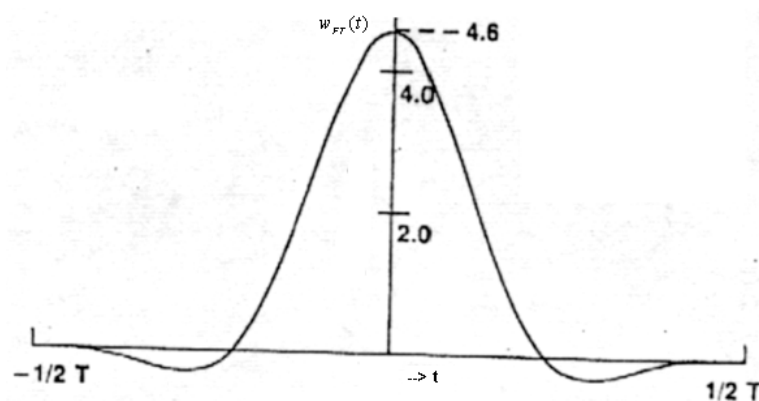


obr.3.7 Spektrum Hanningova okna

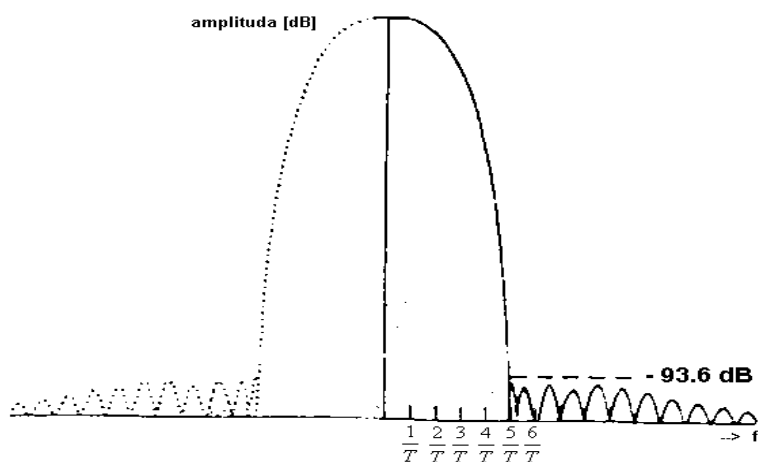
- Okno Flat Top je v intervalu $0 \leq t < T$ definováno vzorcem

$$w_{FT}(t) = 1 - 1.98 \cdot \cos\left(\frac{2 \cdot \pi \cdot t}{T}\right) + 1.29 \cdot \cos\left(\frac{4 \cdot \pi \cdot t}{T}\right) - 0.388 \cdot \cos\left(\frac{6 \cdot \pi \cdot t}{T}\right) +$$

$$+ 0.0322 \cdot \cos\left(\frac{8 \cdot \pi \cdot t}{T}\right) \quad (3.8)$$



obr.3.8 Časový průběh Flat Top okna



obr.3.9 Spektrum Flat Top okna

- Gaussovo okno je vyjádřeno funkcí

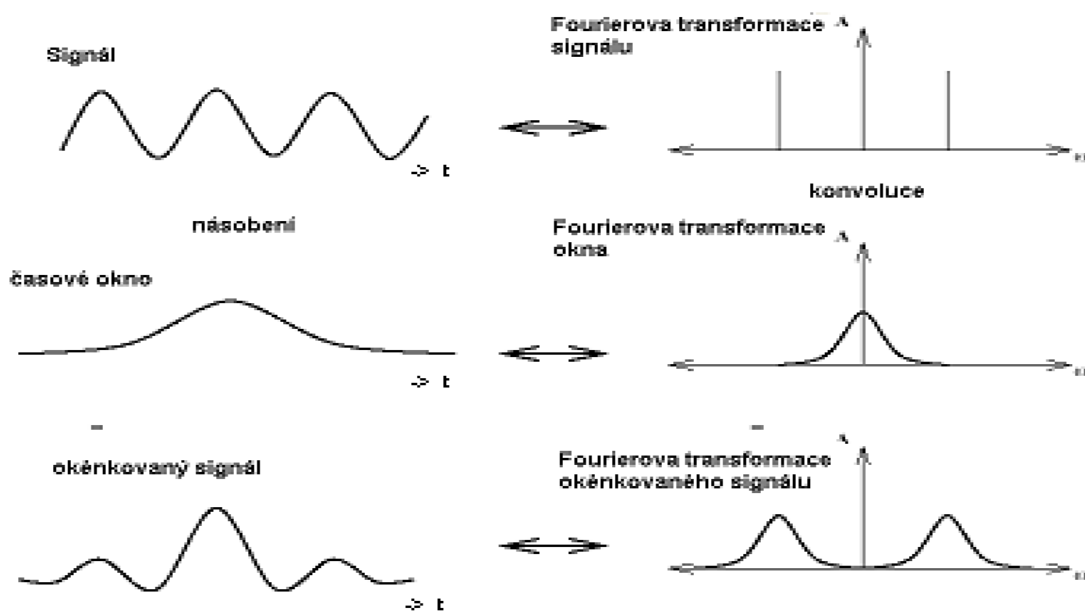
$$w_{GAUSS}(t) = \left(\frac{\alpha}{\pi} \right)^{\frac{1}{4}} \cdot e^{-\frac{\alpha}{2} t^2}, \quad (3.9)$$

kde α je koeficient udávající roztažení Gaussova okna.

Gaussova funkce má tu vlastnost, že se Fourierovou transformací převede na jinou Gaussovu funkci. V logaritmickém amplitudovém měřítku má průběh podobný invertované parabole bez postranních laloků. Z tohoto pohledu se jedná o ideální okno, ale na druhou stranu je šířka pásma Gaussova okna poměrně velká. Protože je Gaussova funkce optimálně soustředěná v časově frekvenční oblasti, z principu neurčitosti vychází

$$\Delta^2 t \cdot \Delta^2 \omega = \frac{1}{\sqrt{\alpha}} \cdot \frac{\sqrt{\alpha}}{2} = \frac{1}{2}, \quad (3.10)$$

což je nejnižší mez neurčitostní nerovnosti, a proto je právě Gaussovo okno silným nástrojem pro STFT. STFT používající Gaussovo okno je známá jako Gaborova transformace.



obr.3.10 Signál vážený Gaussovým oknem

Typ okna	Šířka pásma šumu	Maximální relativní chyba amplitudy(dB)	Nejvyšší postranní lalok (dB)	Pokles laloků (dB/dec)
Obdélníkové	$1,00\Delta f$	3,9	-13,3	20
Hanning	$1,50\Delta f$	1,42	-31,5	60
Flat top	$3,77\Delta f$	0,01	-93,6	0
Gaussovo	$1,9\Delta f$	0,03	nemá	nemá postranní laloky

tab.3.1 Porovnání vlastností časových oken
[4,8,9,12]

3.2. Wiegner-Ville transformace

Alternativní nelineární metodou časově frekvenční analýzy ke krátkodobé Fourierově transformaci pro zpracování stacionárních i nestacionárních signálů je Wiegner Ville transformace (WVT). Wiegnerovo rozdělení bylo v roce 1932 navrženo profesorem Wiegnerem pro oblast kvantové fyziky a zhruba o 15 let později bylo upraveno pro oblast signálové analýzy francouzským vědcem Villem. Wiegner-Ville transformace je definována pro časovou oblast vztahem:

$$WVT(t, f) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) \cdot x^*\left(t - \frac{\tau}{2}\right) \cdot e^{-j2\pi \cdot f \cdot \tau} \cdot d\tau, \quad (3.11)$$

kde $*$ je komplexní konjunkce, t je čas, τ posunutí podél časové osy, x je časová reprezentace signálu a $WVT(t, f)$ je časově frekvenční reprezentace vstupního signálu.

Na rozdíl od metody STFT, u které je rozlišení omezeno okénkovou funkcí, Wiegner-Ville spektrum poskytuje dobré rozlišení jak ve frekvenční, tak i v časové oblasti. Jeho důležitou charakteristikou tedy je, že výpočet není omezen Heisenbergovým principem neurčitosti.

WVT je tedy jednou z nelineárních metod časově frekvenční analýzy, která je založena na zcela odlišném principu než jsou založeny lineární časově frekvenční postupy (např. krátkodobá Fourierova transformace nebo transformace wavelet).

Podotkněme, že všechny lineární časově frekvenční postupy zahrnují principy linearit a superpozice. Pak pro signál $x(t)$ vytvořený lineární kombinací jednotlivých signálových komponent $x_1(t)$ a $x_2(t)$ platí, že časově frekvenční transformace signálu (dále označeno TFR) je lineární kombinací časově frekvenční transformace každé z těchto komponent obsažených v signálu. Tuto skutečnost vyjadřuje rovnice (3.12).

$$x(t) = c_1 \cdot x_1(t) + c_2 \cdot x_2(t) \Rightarrow TFR_x(t, f) = c_1 \cdot TFR_{x_1}(t, f) + c_2 \cdot TFR_{x_2}(t, f) \quad (3.12)$$

Při použití Wiegner-Ville transformace tato rovnice neplatí. Například spektrogram sumy dvou signálů není sumou jednotlivých individuálních spektrogramů a platí pro něj následující vztah

$$\begin{aligned} x(t) &= c_1 \cdot x_1(t) + c_2 \cdot x_2(t) \Rightarrow \\ TFR_x(t, f) &= |c_1|^2 \cdot TFR_{x_1}(t, f) + |c_2|^2 \cdot TFR_{x_2}(t, f) + c_1 \cdot c_2 \cdot TFR_{x_1 x_2}(t, f) + \\ &+ c_2 \cdot c_1 \cdot TFR_{x_2 x_1}(t, f), \end{aligned} \quad (3.13)$$

kde TFR_{x_1} a TFR_{x_2} jsou příspěvky od signálových komponent $x_1(t)$ a $x_2(t)$ a $c_1 \cdot c_2 \cdot TFR_{x_1 x_2}(t, f) + c_2 \cdot c_1 \cdot TFR_{x_2 x_1}(t, f)$ je příspěvek vzniklý interferencí komponent $x_1(t)$ a $x_2(t)$.

Uvedená vlastnost Wiegner-Ville transformace znamená, že analýza složitých signálů nemusí být v určitých případech jednoduchá. Tedy přestože WVT má celou řadu dobrých matematických vlastností, její praktické využití je obtížné. Bylo a je tedy potřebné hledat další nové přístupy, které poskytují kvalitnější resp. jednodušší analýzu signálů.

Nevýhodou WVT je také vznik interferenčních rušivých frekvenčních složek. K jejich odstranění se aplikuje vhodná okénková funkce na rovnici (3.11). Dostáváme tak tzv. pseudo (vyhlazenou) Wiegner-Ville transformaci. Takové řešení poměrně účinně

ně potlačuje interferenci, ale zhoršuje frekvenční rozlišení. Takto upravená transformace sdružuje vlastnosti jak lineárních, tak nelineárních postupů, tedy významně redukuje, při „rozumném“ časovém i frekvenčním rozlišení, vliv interferenčních komponent. Stává se tak dobře použitelným prostředkem k analýze časových signálů [11].

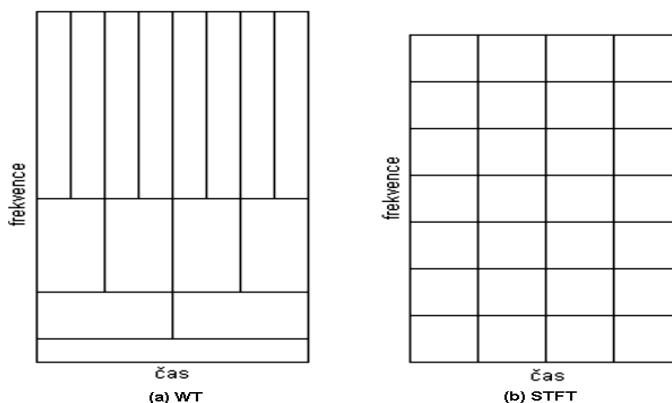
4. Wavelet transformace

Moderní a v poslední době rychle se rozšiřující metodou řešící problémy rozlišení v časové, frekvenční a časově-frekvenční oblasti, je wavelet transformace (WT). Jde o relativně novou metodu vhodnou pro analýzu stacionárních i nestacionárních a rychle se měnících signálů.

WT se rozvinula jako alternativa STFT. Ideou WT je vhodnou změnou šířky okna v čase a jeho tvarem dosáhnout optimálního poměru rozlišitelnosti v čase a frekvenci. Pro nízké frekvence je okno širší, pro vysoké užší. To je hlavní rozdíl od STFT, protože ta používá během celého výpočtu stejně velké časové okno. Díky tomu, že wavelet transformace používá proměnnou velikost okna, mají vyšší frekvence větší rozlišení v čase a nízké jsou zase lépe lokalizovány ve frekvenci. Metodám, které používají různé rozlišení pro různé frekvence se říká analýza s násobným rozlišením [11,13].

4.1. Analýza s násobným rozlišením (MRA)

Analýza s násobným rozlišením (Multiresolution analysis - MRA) analyzuje signál v různých frekvencích s různým rozlišením. MRA je navržena tak, aby pro vysoké frekvence, které většinou trvají krátce, nalezla přesně čas, kdy se v signálu vyskytly. Naopak pro nízké frekvence, které většinou trvají delší dobu, je přesně určena frekvence. Tato vlastnost je vidět na obr.4.1, kde jsou znázorněna šíře oken pro určitá frekvenční spektra wavelet transformace a STFT. Zatímco frekvenční oblast STFT je rozdělena lineárně, frekvenční oblast WT je rozdělena logaritmicky. Obrázek také ilustruje Heisenbergův princip neurčitosti, jehož výsledkem je konstantní velikost plochy obdélníka při jeho různých rozměrech [13,14].



obr.4.1 Časově-kmitočtové rozlišení WT a STFT

4.2. Spojitá waveletová transformace

Spojitá WT (Continuous WT - CWT) byla vyvinuta jako alternativní přístup k STFT za účelem překonání problému s rozlišením. Provádí se jako součin signálu s wavelet funkcí (waveletem), což je jakési okénko, ve kterém je definována funkce s určitými vlastnostmi. V průběhu výpočtu dochází ke změně velikosti okénka a s danou velikostí okénka se vypočtou koeficienty transformace pro každou složku analyzovaného signálu podle vztahu (4.1).

$$CWT_x^\psi(\tau, s) = \Psi_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int_t x(t) \psi^*\left(\frac{t-\tau}{s}\right) dt \quad (4.1)$$

Ze vztahu (4.1) je patrné, že transformovaný signál je funkce dvou proměnných, posunutí τ a měřítka s . $\Psi_x^\psi(\tau, s)$ představuje koeficienty CWT. Člen $\frac{1}{\sqrt{|s|}}$ slouží k energetické normalizaci, takže transformace signálu má stejnou energii v každém měřítku. $x(t)$ je vstupní signál a $\psi(t)$ je mateřský wavelet (mother wavelet – MW). MW je jakýsi prototyp, který generuje další wavelety $\psi\left(\frac{t-\tau}{s}\right)$ zvané dceřinné wavelety, lišící se od MW měřítkem a posunem. Pojem wavelet pak zahrnuje jak MW tak dceřinné wavelety. $\psi^*(t)$ značí komplexně sdružené číslo k $\psi(t)$.

Pojem časové posunutí τ je použit ve stejném významu jako u STFT. Jedná se o posun waveletu vzhledem k signálu.

Měřítka s je velice podobné měřítku v mapách. Velké měřítko znamená pohled z velké výšky bez detailů, a malé měřítko naopak pohled zblízka s podrobnými detaily. Proto velké měřítko odpovídá nízkým frekvencím a popisuje signál z globálního hlediska (nízké frekvence často leží v celém signálu) a malé měřítko odpovídá vysokým frekvencím, které popisují detaily signálu (většinou trvají relativně krátkou dobu). Je tedy jasné, že mezi frekvencí a měřítkem bude platit vztah:

$$s \approx \frac{1}{f} \quad (4.2)$$

Kompresi časového měřítka z $s(t)$ na $s(2t)$ odpovídá ve Fourierově transformaci operaci

$$s(t) \Leftrightarrow S(f), \quad s(2t) \Leftrightarrow \frac{1}{2} S\left(\frac{f}{2}\right) \quad (4.3)$$

Když je časové měřítko komprimováno 2, pak se kmitočtové pásmo roztáhne o oktávu výš (násobeno 2).

Jestliže pouze chceme analyzovat signál a nechceme poté zpětně zrekonstruovat původní signál, pak mateřský wavelet $\psi(t)$ může být jakákoliv funkce, která má nulovou střední hodnotu, to znamená, že musí alespoň jednou kmitnout:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0. \quad (4.4)$$

Ovšem když je vyžadována i perfektní zpětná rekonstrukce signálu, výběr MW je více omezený. Musí splňovat podmínku danou vzorcem (4.5)

$$C_{\psi} = \frac{1}{2\pi} \int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty, \quad (4.5)$$

kde $\Psi(\omega)$ je Fourierova transformace mateřského waveletu $\psi(t)$. Podmínka (4.5) předpokládá $\Psi(0) = 0$ a říká, že wavelet má konečnou délku, či v $\pm \infty$ dosahuje nulové hodnoty. Jinými slovy, mateřský wavelet je pásmová propust. Jakmile $\psi(t)$ vyhovuje této podmínce, pak původní signál $x(t)$ může být zrekonstruován vzorcem (4.6) [1,12,13,14]

$$x(t) = \frac{1}{C_{\psi}} \int_s \int_{\tau} CWT_x^{\psi}(\tau, s) \frac{1}{s^2} \psi\left(\frac{t-\tau}{s}\right) d\tau ds. \quad (4.6)$$

4.2.1. Vlastnosti CWT

- Linearita

$$(CWT(ax_1 + bx_2))(\tau, s) = a(CWT(x_1))(\tau, s) + b(CWT(x_2))(\tau, s) \quad (4.7)$$

- Invariance v čase

$$CWT(x_2)(\tau, s) = CWT(x_1)(\tau - b, s), \quad x_2(t + b) = x_1(t) \quad (4.8)$$

Invariance v čase popisuje skutečnost, že posun analyzované funkce po časové ose způsobí stejný posun wavelet koeficientů po ose polohy.

- Dilatace

$$CWT(x_2)(\tau, s) = CWT(x_1)\left(a\tau, \frac{s}{a}\right), \quad x_2 = \sqrt{|a|} \cdot x_1(at), \quad a \neq 0, \quad (4.9)$$

Vztah popisuje závislost mezi CWT originální funkcí a její roztaženou nebo zúženou podobou, ve wavelet koeficientech dojde k adekvátnímu roztažení v ose polohy a k posunu v ose měřítka [15].

4.2.2. Výpočet CWT

Pro výpočet CWT koeficientů je třeba vybrat si MW z předem definovaných (např. viz. kapitola 4.5) nebo je možné vytvořit si vlastní.

Vezmeme wavelet o základní velikosti, která je reprezentována koeficientem $s = 1$. Umístíme ho na začátek signálu a vynásobíme je spolu. Signál v místech kam wavelet nezasáhne bude nulový. Wavelet je jakási obdoba okénka, jen s tím rozdílem, že na okénko se musela ještě aplikovat některá z Fourierových metod např. FFT, zatímco u wavelet transformace dostaneme amplitudu pouhým sečtením všech hodnot vynásobeného signálu. Tuto sumu ještě znormujeme vynásobením konstantou $\frac{1}{\sqrt{s}}$, aby měl transformovaný signál pro všechny měřítka stejnou energii. Dostali jsme tedy koeficient wavelet transformace pro měřítko $s = 1$ a posunutí $\tau = 0$.

Pokud bude mít signál zrovna v počítané části tvar odpovídající tvaru waveletu, bude mít koeficient relativně velkou hodnotu. Pokud však nebude signál waveletu příliš odpovídat, koeficient bude malý nebo dokonce nulový.

Nyní posuneme wavelet v signálu na pozici τ a vypočteme koeficient wavelet transformace pro $s = 1$ a $\tau = 1$. Takto se postupuje dokud se nedostaneme s waveletem až na konec signálu.

Potom je měřítko s zvětšeno o malou hodnotu (protože se jedná o CWT musí být jak s tak τ zvětšováno spojitě), tím se wavelet roztáhne. Umístíme ho opět na začátek signálu, vynásobíme a znormujeme. Protože se změnilo měřítko s , normujeme už jinou hodnotou, což je v pořádku, jelikož jsme wavelet zvětšili a musíme ho tedy dělit také větším číslem. Takto se postupuje dál a dál, pro všechny hodnoty měřítka s , vždy pro celý signál.

Při výpočtu koeficientů jsme začali s měřítkem $s = 1$ a pak jsme wavelet stále zvětšovali. Tím jsme vlastně v dané části signálu vždy hledali frekvenci odpovídající waveletu. Na začátku je s minimální, ze vztahu (4.2) mezi měřítkem a frekvencí tedy vyplývá, že hledáme maximální frekvenci. Jak bylo uvedeno výše, wavelet transformace provádí změnu velikosti okna podle roztažení waveletu, a tím získává pro vysoké

frekvence (resp. pro malé měřítko) přesné určení v čase a naopak. Na postupu výpočtu je tento fakt vidět také. Pokud máme malý wavelet (což na začátku máme), vejde se nám do celého signálu vícekrát než wavelet z konce výpočtu, kde je měřítko s velké [12,13,14].

4.3. Diskrétní waveletová transformace

Za účelem zjednodušení výpočtu koeficientů WT se začala využívat diskrétní forma wavelet transformace (Discrete Wavelet Transform - DWT), která poskytuje dostatečné informace pro analýzu i syntézu originálního signálu.

Dvojkovou závislostí parametrů s (měřítko) a τ (posun) můžeme vytvořit z vhodného waveletu ψ ortonormální bázi:

$$s = 2^p, \quad \tau = 2^p \cdot k, \quad p, k \in \mathbb{Z} \quad (4.10)$$

pak

$$\Psi_{k,p}(t) = \frac{1}{\sqrt{2^p}} \psi\left(\frac{t - 2^p \cdot k}{2^p}\right), \quad (4.11)$$

kde p odpovídá měřítku, k poloze. Tento postup mění měřítko s na násobky dvou, a proto bývá nazýván dyadickou diskrétní wavelet transformací. Díky ortonormalitě takto volený wavelet umožňuje neredundantní dekompozici signálu [13,14].

4.3.1. Číslicové filtry

Číslicový filtr je algoritmus nebo obvod, který požadovaným způsobem mění spektrum vstupního diskrétního signálu. Číslicové filtry jsou někdy považovány za třetí generaci filtrů, následující po analogových pasivních filtrech a analogových aktivních filtrech využívajících operační zesilovače. Je možné je realizovat např. pomocí speciálních mikroprocesorů a mikropočítačů s příslušnými programy, pomocí speciálních mikropočítačů – tzv. číslicových signálových procesorů, nebo pomocí speciálních jednoúčelových obvodů.

Velkou výhodou číslicových filtrů je možnost změny parametrů filtru změnou hodnot koeficientů filtru. Změnou parametrů filtru lze pro stejný hardware změnit i typ filtru. Navíc je možné diskrétní signál z výstupu filtru opakovaně přivést na vstup a tak zvýšit řád filtru.

Číslicové filtry se dělí podle délky impulzní odezvy na filtry typu FIR (Finite Impulse Response), jejichž impulzní odezva $h[n]$ má konečný počet členů, a filtry typu IIR (Infinite Impulse Response), jejichž impulzní odezva je časově neomezená. Filtry typu IIR se pro wavelet transformaci nepoužívají, proto se dále omezíme pouze na popis číslicových filtrů typu FIR.

Přenosová funkce $H(z)$ má tvar vztahu (4.12).

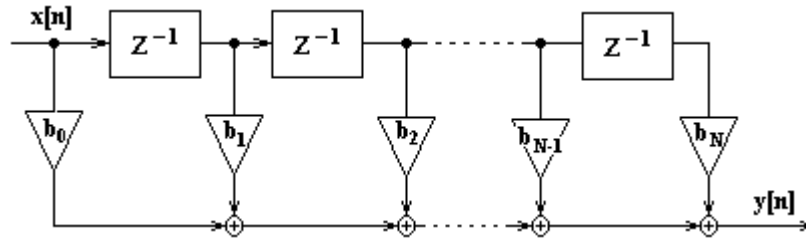
$$H(z) = \sum_{m=0}^{N-1} b_m \cdot z^{-m} = \sum_{n=0}^{N-1} h[n] \cdot z^{-n} \quad (4.12)$$

Diferenční rovnice má tvar konvoluce koeficientů b_m s historií vstupu:

$$y[n] = \sum_{m=0}^{N-1} b_m \cdot x[n-m], \quad (4.13)$$

čili hodnota výstupního signálu $y[n]$ závisí na současné hodnotě vstupu a $N+1$ předchozích hodnotách vstupu.

Struktura k diferenční rovnici (4.13) je tzv. transversální filtr:



obr.4.2 Přímá struktura FIR filtru – transversální filtr

Je zřejmé, že koeficienty násobiček transversálního filtru jsou přímo hodnoty impulzní odezvy filtru, čili

$$b_m = h[m] \quad (4.14)$$

Ustálený stav při filtraci vstupního signálu $x[n]$ nastane až po naplnění všech zpoždovacích členů, tj. po $N+1$ taktech.

Frekvenční odezvu filtru najdeme jako $H[z]$, pro $z=e^{j\omega}$:

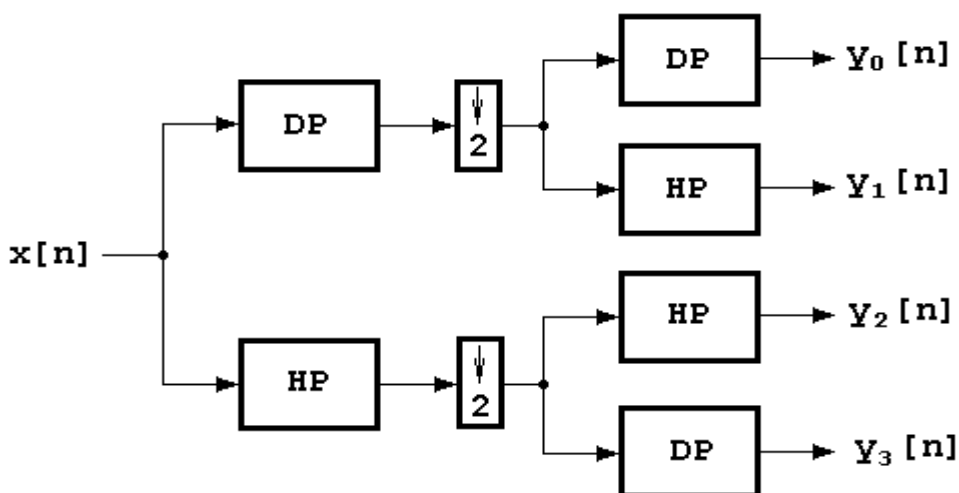
$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h[n] \cdot e^{-jn\omega} \quad (4.15)$$

Číslicové filtry typu FIR mají výhody v absolutní stabilitě a linearitě fázové kmitočtové charakteristiky v celém kmitočtovém rozsahu. Jejich realizace je nerekurzivní, a tudíž struktury neobsahují zpětné vazby. Provedení FIR filtrů ovšem vyžaduje velkou paměť pro ukládání koeficientů a stavových proměnných. Nevýhodou je také jejich nižší výkon ve filtraci. Tím se má na mysli, že při stejné délce filtru, tj. počtu koeficientů, nebude strmost FIR filtru nikdy dosahovat hodnot filtru IIR ekvivalentní délky [1,10,16].

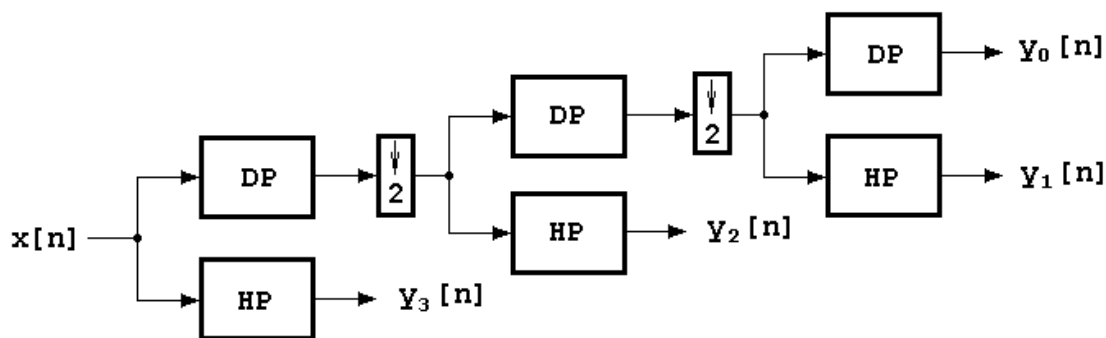
4.3.2. Banky filtrů

Zpracování signálu diskrétní wavelet transformací probíhá prostřednictvím banky filtrů. Banku filtrů tvoří spojení několika číslicových (FIR) filtrů do skupiny. Banka filtrů v základním zapojení se skládá ze dvou úseků.

V prvním úseku jsou filtry, které provádí analýzu signálu tím, že rozdělí kmitočtové pásmo pomocí dolní a horní propusti na dvě rovnoměrné části. Spektrum signálu je rozděleno na nízkofrekvenční část a vysokofrekvenční část. Toto dělení může dále pokračovat členěním do dalších subpásem opět pomocí dolní a horní propusti. Není ovšem nutné zachovávat celou délku výstupních signálů analyzujících filtrů (všechny jejich hodnoty), neboť filtrací zmenšujeme velikost kmitočtového pásma. Proto můžeme postupně snižovat vzorkovací kmitočet tím, že vynecháme například sudě označené složky subsignálu a ponecháme pouze složky s lichými indexy. Tento postup se nazývá decimace s činitelem 2.



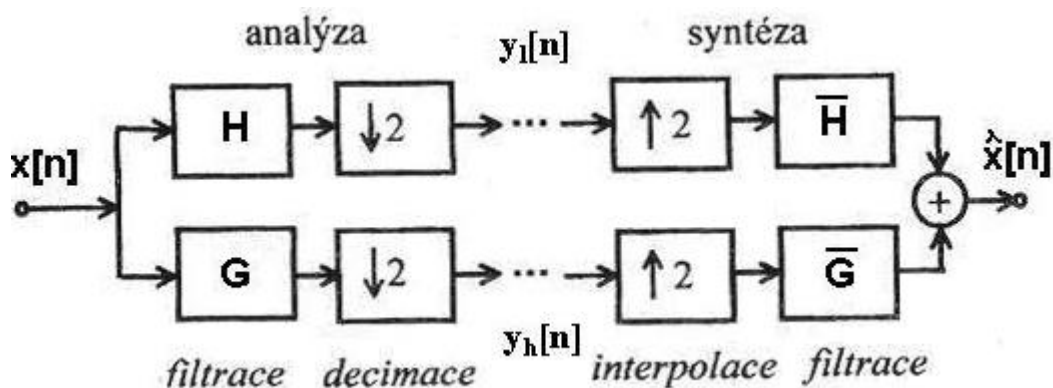
obr.4.3 Stromová struktura vytvářející lineární banku filtrů



obr.4.4 Kaskádní struktura vytvářející exponenciální banku filtrů

Druhým úsekem banky filtrů je sekce syntézy. Subsignály jsou interpolovány s činitelem 2, kdy za každým vzorkem původní sekvence následuje doplněný vzorek (buďto nulová hodnota nebo interpolační hodnota mezi dvěma následujícími vzorky) a je prováděna opět filtrace dvojicemi filtrů typu horní a dolní propusti. Tím postupně zvyšujeme hodnotu vzorkovacího kmitočtu.

Takto lze získat takové vlastnosti zpracování signálů, které se samostatnými filtry nedají dosáhnout. Na obr.4.5 je znázorněno základní schéma banky filtrů pro analýzu a syntézu.



obr.4.5 Základní blokové schéma dvoukanálové banky filtrů

Přenosové funkce H a \bar{H} patří dolním propustem a přenosové funkce G a \bar{G} odpovídají horním propustem.

Předpokládáme, že všechny použité filtry jsou lineární a časově invariantní. Aby se filtr například při pětinasobném opakujícím se snižování vzorkovacího kmitočtu choval dobře a aby banka FIR filtrů splňovala podmínku perfektní rekonstrukce (bezeztrátové komprese), pak musí platit dvě podmínky:

- Nesmí dojít k prosakování (tj. k překrývání sousedních spekter a prosakování složek ze základního pásma do všech ostatních vlivem periodičnosti kmitočtové charakteristiky).
- Nesmí docházet ke zkreslení (může pouze nastat zpoždění signálu po zpracování ve tvaru $\hat{x}[n] = x[n - k]$. O kolik vzorkovacích intervalů k bude vstupní signál zpožděn záleží na řádu filtrů používaných v bance filtrů) [1,10].

4.3.3. Výpočet DWT

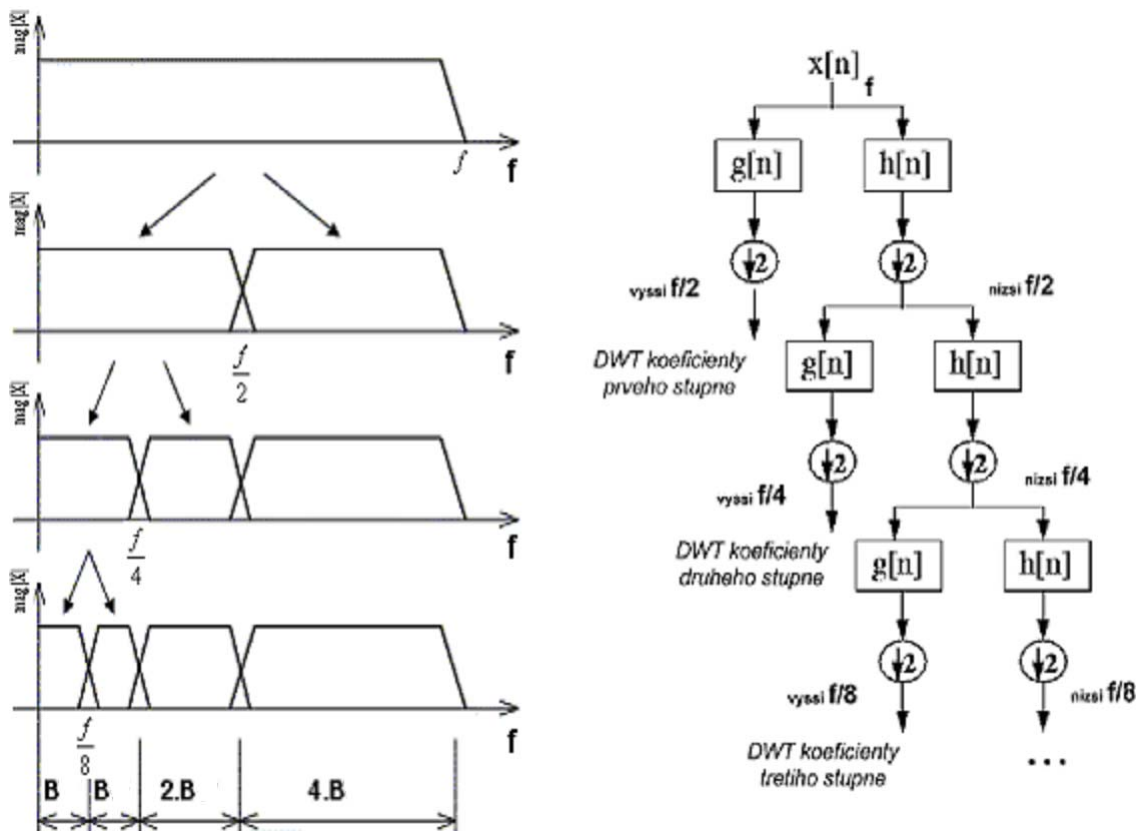
Nyní bude podrobněji popsána procedura výpočtu jednoho kroku analýzy signálu pomocí DWT na dolnoproputném filtru. Procedura začíná průchodem signálu dolní propustí s prahem v polovině frekvenčního pásma a impulsní odezvou $h[n]$. Matematicky lze funkci filtru vyjádřit vztahem (4.16).

$$x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k] \cdot h[n - k] \quad (4.16)$$

Tento filtr odstraní všechny frekvence vyšší než je polovina nejvyšší frekvence v signálu, ale nezmění počet vzorků v signálu. Ten je změněn až podvzorkováním (decimací), kdy je odstraněn každý druhý vzorek ze signálu. Toto lze zapsat vztahem (4.17).

$$x[n] * h[n] = \sum_{k=-\infty}^{+\infty} h[k] \cdot x[2n - k] \quad (4.17)$$

Další stupně dekompozice signálu se potom provedou pouze na signálu, který prošel dolnoproputným filtrem. Operace podvzorkování se provede také se signálem u hornoproputného filtru s impulsní odezvou $g[n]$ (viz. obr.4.6).



obr.4.6 Frekvenční pohled na diskrétní wavelet transformaci

Oba filtry, dolní propust h a horní propust g , tvoří pár zrcadlových filtrů, které mají komplementární propustná pásma. Výsledkem filtrace jsou dva nové diskrétní signály, které jsou označovány jako aproximační a detailní složka. Tyto názvy vycházejí z předpokladu, že každý signál má základní informace, které postačí k aproximaci daného signálu, uloženy v nízkofrekvenčním pásmu. Vysokofrekvenční pásmo obsahuje informace, které zpřesňují aproximaci, tj. detaily průběhu signálu.

Zatímco každá filtrace odstraňuje část informace ze signálu, decimací se odstraní pouze redundance. Z původního signálu jsou odstraněny filtrací všechny frekvence vyšší než je polovina Nyquistovi frekvence, a proto pro další zpracování signálu již stačí k vyjádření nejvyšší frekvence polovina vzorků.

DWT tedy rozkládá signál do různých frekvenčních oblastí s různým rozlišením dekomponovaného signálu. Rozklad signálu na rozdílná frekvenční pásma získáme postupným stupňováním dekompozice vstupního signálu. Ze vstupního signálu získáme při prvním stupni posloupnost $y_h[n]$ (u hornopropustného filtru) a $y_l[n]$ (u dolnopropustného filtru). Matematicky lze tento první krok rozkladu signálu popsat vztahem (4.18) pro posloupnost $y_h[n]$ a vztahem (4.19) pro posloupnost $y_l[n]$.

$$y_h[n] = \sum_k x[k] \cdot g[2n - k] \quad (4.18)$$

$$y_l[n] = \sum_k x[k] \cdot h[2n - k] \quad (4.19)$$

Posloupnost $y_h[n]$ nazveme DWT koeficienty. V případě dekompozice n -tého stupně signálu jsou DWT koeficienty tvořeny všemi posloupnostmi $y_h[n]$. Tento postup rozkladu signálu je znázorněn na obrázku 4.6 vpravo a je nazýván Mallatovým algoritmem. Dekompozice signálu může probíhat až do doby, než je k dispozici pouze jedna hodnota v posloupnosti $y_l[n]$. S klesajícím kmitočtem klesá také šířka pásma (B) filtru.

Protože je počet koeficientů shodný s počtem vzorků a nedochází ke ztrátě informace, popis signálu je neredundantní. Poněvadž je popis také úplný, pomocí inverzního postupu k postupu na obrázku 4.6 lze přesně rekonstruovat analyzovaný signál. Inverzní diskrétní wavelet transformace se označuje IDWT. Operaci podvzorkování nahradíme operací převzorkování (interpolace). Místo analyzujících filtrů je použita další dvojice tentokrát rekonstrukčních filtrů \bar{h} a \bar{g} . Této čtveřici filtrů se říká kvadraturní zrcadlové filtry a jejich tvar je přesně dán typem waveletu. Při IDWT se v pyramidě na obr.4.6 vpravo postupuje od listů nahoru ke kořenu [13,14,15].

4.3.4. Wavelet pakety

Rozšířením rozkladového dyadického algoritmu DWT (obr.4.6) je rozkladové schéma tzv. wavelet paketů. Paketový rozklad signálu se od dyadického liší v tom, že jsou postupně rozkládána všechna filtrovaná pásma (viz. obr.4.3). To vede k rovnoměrnému rozdělení frekvenční osy na rozdíl od oktávového rozdělení v případě dyadického rozkladu.

V případě dyadické DWT je strom rozkladu vždy jednoznačně určen – rozkládá se vždy jen pásmo aproximačních koeficientů. Paketový rozklad umožňuje dále analyzovat i detailní složky a zpřesňovat tak frekvenční lokalizaci výsledných koeficientů. Tento postup vede k zahuštění rozkladových frekvencí a je tedy redundantní [17].

4.4. Wavelet funkce

Wavelet funkce jsou získány na základě řešení měřítkových rovnic. Obecná měřítková rovnice N -tého řádu má tvar:

$$\phi(t) = \sqrt{2} \cdot \sum_{n=0}^{N-1} h(n) \cdot \phi(2t - n) \quad (4.20)$$

Řešením rovnice (4.20) dostaneme měřítkovou funkci (scaling function) $\phi(t)$ a pomocí ní je možné vytvořit wavelet rovnici:

$$\psi(t) = \sqrt{2} \cdot \sum_{n=0}^{N-1} g(n) \cdot \phi(2t - n) \quad (4.21)$$

Je vhodné normalizovat plochu pod funkcí měřítka $\phi(t)$, tak aby byla rovna jedné, tj.

$$\int_{-\infty}^{\infty} \phi(t) dt = 1, \quad (4.22)$$

potom pro součet koeficientů impulsní charakteristiky platí [1,12]:

$$\sum_{n=0}^{N-1} h(n) = \sqrt{2}, \quad (4.23)$$

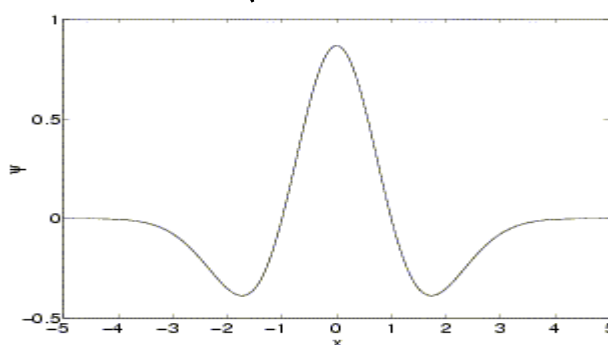
$$\sum_{n=0}^{N-1} g(n) = 0 \quad (4.24)$$

4.5. Některé používané wavelety

Existuje kolem 400 používaných waveletů, které jsou více či méně vhodné pro různé úlohy. Zde bude uvedeno několik nejznámějších spolu s jejich základními vlastnostmi.

- Wavelet Mexican hat
 - má tvar druhé derivace průběhu hustoty pravděpodobnosti Gaussova rozdělení
 - vlastnosti: symetrický, vhodný pro CWT, není ortogonální (nelze použít pro diskrétní waveletovou transformaci - DWT)

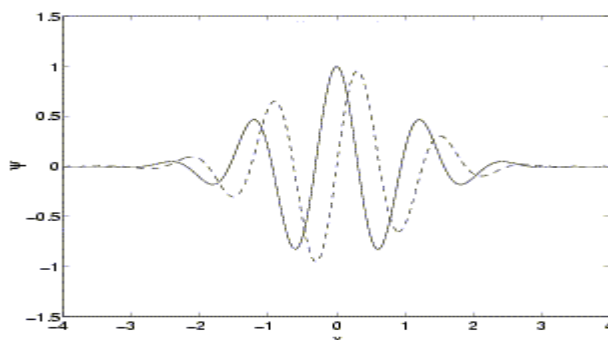
$$\psi(x) = \frac{2}{\sqrt{3}} \pi^{\frac{1}{4}} (1 - 2x^2) e^{-x^2} \quad (4.25)$$



obr.4.7 Mexican hat

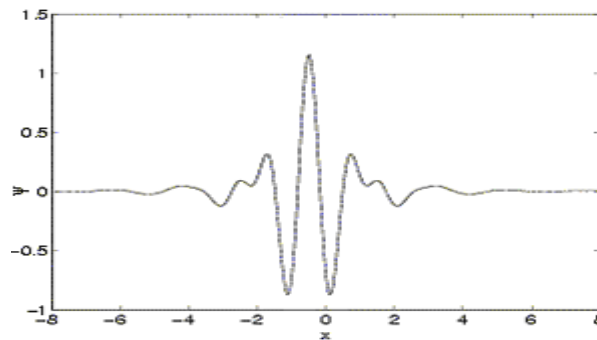
- Morletův wavelet
 - má tvar komplexní sinusovky modulované Gaussovským oknem
 - jedná se o kompromis mezi přesností polohy (lepší je např. Mexikan hat) a frekvenčním rozlišením
 - vlastnosti: symetrický, komplexní, vhodný pro CWT, není ortogonální

$$\psi(x) = a \cdot e^{-\frac{1}{2}x^2} (\cos(5x) + j \sin(5x)) \quad (4.26)$$

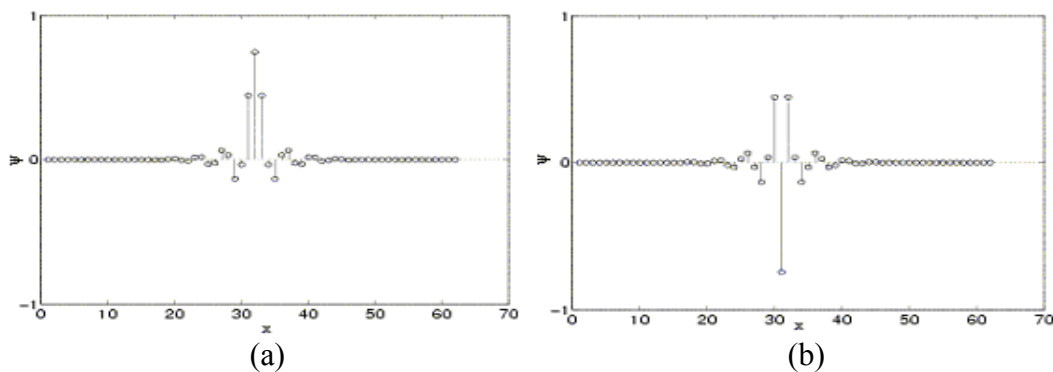


obr.4.8 Morletův wavelet

- Mayerův wavelet
 - je definován ve frekvenční doméně, nemá explicitní vzorec pro vyjádření v čase
 - v originálním tvaru (viz. obrázek 4.9) nelze realizovat FIR filtry pro DWT a proto se vytvořila jeho diskrétní aproximace (viz .obrázek 4.10)
 - vlastnosti: symetrický, ortogonální, je vhodný pro CWT i DWT



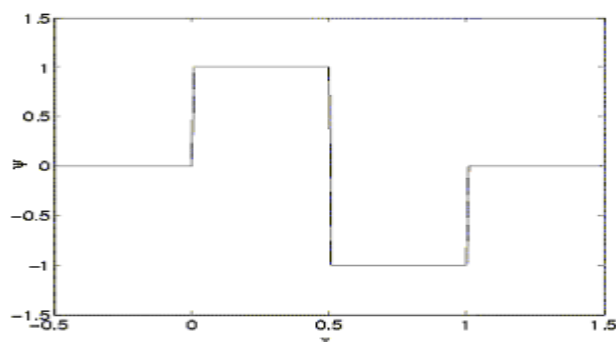
obr.4.9 Mayerův wavelet



obr.4.10 Konvoluční filtry pro výpočet DWT pomocí Mayerova waveletu
a) dolní propust, b) horní propust

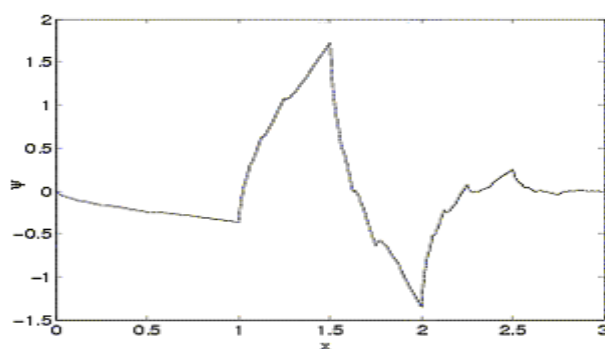
- Haarův wavelet
 - jednoduchý wavelet, který ale neumožňuje hladkou rekonstrukci signálu
 - vlastnosti: symetrický, ortogonální, vhodný pro CWT i DWT, jednoduchá implementace, je nespojitý (to je i přes ostatní výhody velké mínus)

$$\psi(x) = \begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & \text{pro } 1/2 < x < 1 \\ 0 & \text{jinde} \end{cases} \quad (4.27)$$



obr.4.11 Haarův wavelet

- Wavelet Daubechies
 - představuje skupinu waveletů různého řádu $n \geq 1$ ($n = 1$ Haarův wavelet)
 - nemají (kromě Haarova) explicitní vyjádření waveletové funkce
 - vlastnosti: asymetrický (kromě 1. řádu), ortogonální, vhodný pro CWT i DWT [12,13,14]



obr.4.12 Wavelet Daubichies

4.5.1. Časově frekvenční rozlišení wavelet funkcí

Časové a frekvenční rozlišení různých wavelet funkcí se liší. Ideální hodnota rozlišení je reprezentována rovností $\Delta^2 t \cdot \Delta^2 \omega = 0.5$. Výsledky pro vybrané wavelet funkce jsou uvedeny v tabulce 4.1 [18].

wavelet	$\Delta^2 t$	$\Delta^2 \omega$	$\Delta^2 t \Delta^2 \omega$
Morlet	0,7071	0,7081	0,5007
Meyer	0,8418	0,9824	0,8271
Daubichies řádu 2	1,540	9,424	14,51
Haar	0,5775	130,67	75,44

tab.4.1 Časové, frekvenční a časově frekvenční rozlišení vybraných wavelet funkcí

4.5.2. Výběr waveletu

Výběr vhodného waveletu záleží vždy na konkrétní úloze. Většinou se volí na základě pokusů a omylů. Bylo ale objeveno několik souvislostí, které mohou při výběru pomoci:

- Komplexní wavelety jako Morletův detekují dobře oscilace, nejsou vhodné pro detekci osamocených singularit.
- K detekci špiček a singularit v signálu jsou vhodné čistě reálné wavelety s málo oscilacemi.
- Symetrické wavelety nezpůsobují fázový posuv mezi špičkou, singularitou, oscilací v signálu a polohou příslušného koeficientu při transformaci.
- Abychom detekovali současně amplitudu a fázi, musíme použít komplexní wavelet (Morletův) [14].

5. Číslicové signálové procesory pro efektivní zpracování signálů

Zpracování signálů vyžaduje rychlé výpočty komplexních a opakujících se algoritmů a mnoho aplikací potřebuje provádět výpočty v reálném čase. Signál je spojitá funkce času, která musí být pro číslicové zpracování navzorkována a převedena do digitální podoby. Realizace výpočetního algoritmu na procesoru musí tedy být kratší, než je časový interval mezi dvěma po sobě následujícími vzorky signálu, který je zpracováván.

Architektura číslicového signálového procesoru (Digital Signal Processor - DSP) je optimalizována pro vykonávání takových algoritmů. Charakteristiky dobrého signálového procesoru zahrnují: rychlé a flexibilní aritmetické výpočetní jednotky (násobičky, střadače, posouvače); neomezený datový tok dovnitř a ven z výpočetních jednotek; rozšířenou přesnost a dynamický rozsah ve výpočetních jednotkách (aby nedošlo k přetečení a aby se minimalizovaly zaokrouhlovací chyby); dvojí generátory adres (pro současnou obsluhu dvou vstupů při duálních operacích); účinné sekvenční zpracování programu (zahrnující schopnost efektivně pracovat se smyčkami a přerušeními) a pohodlí programování.

Číslicové signálové procesory lze rozdělit do dvou základních tříd. DSP s pevnou řádovou čárkou mají typicky 16ti bitovou architekturu založenou na datech typu integer. A DSP s plovoucí řádovou čárkou mají obvykle 32bitovou architekturu, založenou na datovém typu float obsahující mantisu a exponent [19].

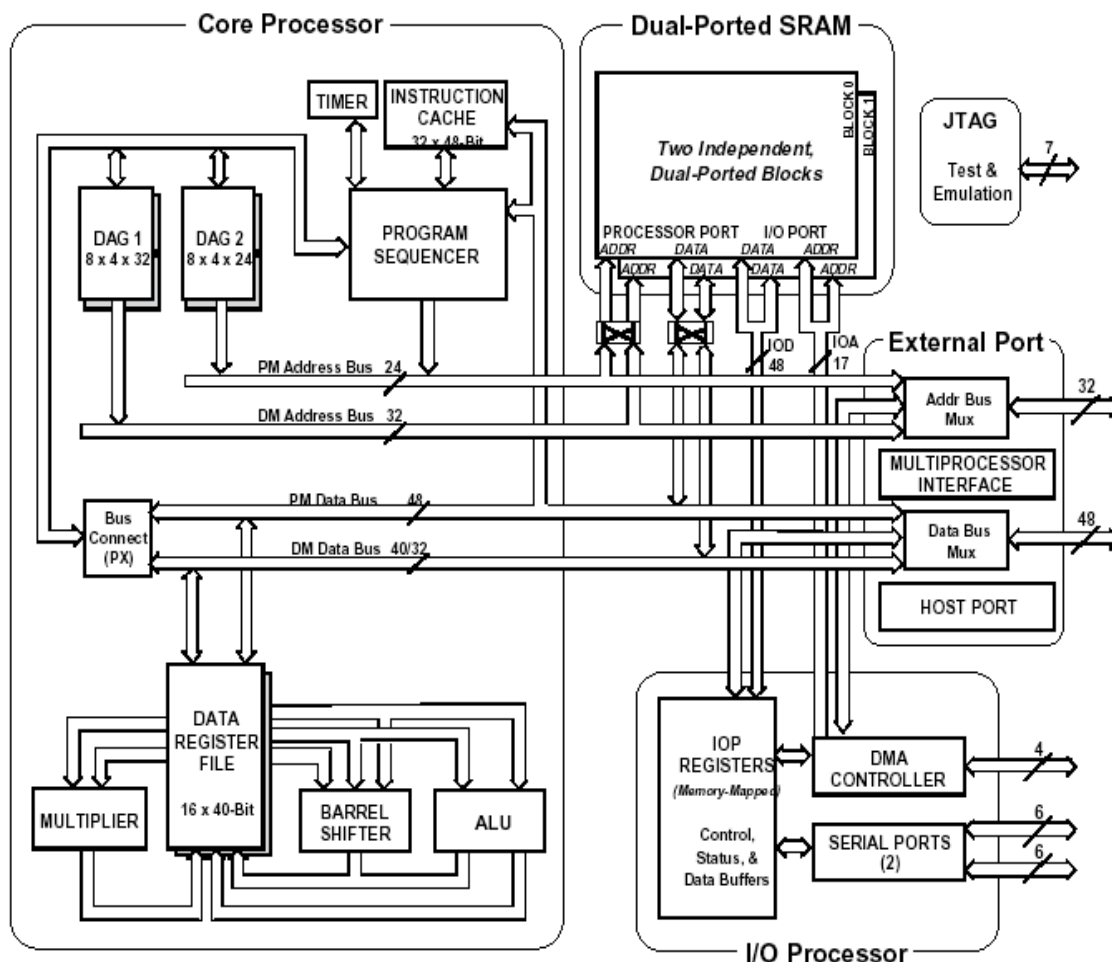
5.1. Signálový procesor ADSP21061

Realizace wavelet analyzátoru v této práci předpokládá použití vývojové desky EZ-KIT, která je osazena procesorem ADSP21061. Jako příklad za všechny signálové procesory proto podrobněji popíšeme architekturu právě tohoto DSP.

SHARC (Super Harvard Architecture Computer) ADSP21061 je vysoce výkonný DSP vhodný do aplikací pro zpracování řeči, zvuku a obrazu.

Všechny záležitosti kolem architektury, detailního popisu bloků, programování, obsluhy programového vybavení, komunikace jsou důkladně popsány v dokumentaci AnalogDevices v [20]. Dále proto bude uveden jen nejnutnější popis jádra procesoru s jeho implementovanými prostředky a možnostmi.

Výpočetní schopnosti procesoru ADSP21061 se odvíjejí od jeho vnitřní struktury (viz. obr.5.1) [10,20].



obr.5.1 Blokové schéma vnitřní struktury procesoru ADSP21061

5.1.1. Základní struktura

Architektura procesoru ADSP21061 obsahuje následující rysy:

- Nezávislé, paralelní výpočetní jednotky

Aritmeticko logická jednotka (ALU), násobička (Multiplier), a posouvač (Barel Shifter), všechny vykonávají instrukce v jediném cyklu. Tyto tři jednotky jsou uspořádány paralelně aby maximalizovali výpočetní výkonnost. Paralelní operace ALU a násobičky tvoří samotné instrukce vykonávané během jednoho taktu. Výpočetní jednotky podporují IEEE základní 32-bitový, rozšířený 40-bitový datový formát v plovoucí řádové čárce a 32-bitový datový formát v čárce pevné.

- Soubor datových registrů (Data Register File)

Hlavní účel souboru datových registrů je přenášet data mezi výpočetními jednotkami a datovými sběrnicemi a uchovávat mezivýsledky. Tento 10-portový, 32-registrový soubor registrů umožňuje stálý datový tok mezi výpočetními jednotkami a vnitřní pamětí.

- PX registry připojení sběrnice (Buss Connect)

Registry PX umožňují přenos dat mezi 48-bitovou PM datovou sběrnicí a 40-bitovou DM datovou sběrnicí nebo mezi 40-bitovým souborem datových registrů a PM datovou sběrnicí. Tyto registry obsahují hardware na zvládnutí 8-bitové difference.

- Současné získání instrukce a dvou operandů v jednom cyklu

ADSP21061 má rozšířenou Harvardskou architekturu, ve které sběrnice datové paměti (DM Data Bus) přenáší data a sběrnice programové paměti (PM Data Bus) přenáší instrukce a data. S oddělenou programovou a datovou sběrnicí a cache pamětí pro instrukce (Instruction Cache) procesor může v jednom cyklu současně získat dva operandy a instrukci (z cache).

- Instrukční cache

Instrukční cache umožňuje operace získávání instrukce a dvou datových hodnot současně. Na cache jsou ukládány pouze instrukce, ke kterým se nedá přistupovat pomocí PM datové sběrnice.

- Generátory datových adres (DAG1 a DAG2) s hardwarovými kruhovými buffery

Dva nezávislé adresové generátory DAG1 a DAG2 (Data Address Generator) vytvářejí adresy dvou operandů pro jejich současný výběr z datové a programové paměti. DAG1 pracuje s adresovou sběrnicí paměti dat (32-bitová adresa) a DAG2 slouží pro adresovou sběrnici paměti programu (24 bitová adresa). Generátory DAG1 a DAG2 umožňují, aby byl obsah adresových ukazatelů změněn o libovolnou hodnotu po každém přístupu do paměti programu i dat nepřímým adresováním. To znamená, že instrukce určuje DAG registr, který obsahuje adresu, namísto přímé adresové hodnoty.

Generátory datových adres mají též implementovány hardwarové kruhové buffery, které umožňují efektivní programování zpožďovacích linek a dalších datových struktur požadovaných při digitálním zpracování signálu, a obvykle se používají při digitální filtraci a Fourierově transformaci. Tyto generátory adres obsahují dostaču-

jící počet registrů k vytvoření více než 32 kruhových bufferů. DAGy automaticky ovládají cyklické přetáčení adresových pointerů a tím zjednodušují implementaci. Kruhové buffery mohou začít a skončit na libovolné poloze v paměti.

- Flexibilní sada instrukcí

48-bitové instrukční slovo vyhovuje různorodým paralelním operacím pro stručné a zhuštěné programování. Například procesor ADSP21061 může provádět násobení, sčítání, odčítání a dělení v jediné komplexní instrukci, která je vykonávána v jednom taktu.

- Duální paměť (Dual Ported SRAM)

Procesor obsahuje 1 megabit interní dvouportové paměti SRAM na čipu rozdělené do dvou stejně velkých bloků. Každý blok má osm 16-bitových sloupců s 4k 16-bitových slov na sloupec. Každý blok paměti je dvouportový a to dovoluje nezávislý přístup jádra procesoru a I/O procesoru nebo DMA řadiče v jednom cyklu. Dvouportová paměť a oddělené sběrnice na čipu umožňují dva datové přenosy z jádra a jeden z I/O v jediném cyklu. Paměť může být konfigurovatelná jako 32k slov 32-bitových dat, 64k slov 16-bitových dat, 16k slov 48-bitových instrukcí nebo různé kombinace do maximální velikosti 1Mb.

Ačkoliv každý blok dat může uchovávat kombinaci instrukcí a dat, neefektivnější přístup k paměti je, když jeden blok uchovává pouze data a k přenosu využívá DM datovou sběrnici, a druhý blok uchovává instrukce i data a k přenosu využívá PM datovou sběrnici. Používání DM a PM datové sběrnice v tomto smyslu, zajišťuje přenos dvou datových hodnot v jednom cyklu. V tomto případě musí být instrukce k dispozici v cache.

- DMA řadič (DMA Controller)

DMA řadič umožňuje nerušený datový přenos bez zásahu procesoru. DMA řadič pracuje nezávisle na jádru procesoru, což umožňuje DMA operace probíhající současně s vykonáváním programových instrukcí jádrem procesoru. DMA přenos může nastat mezi vnitřní pamětí procesoru a externí pamětí, externími periferiemi, sériovými porty nebo hostitelským procesorem. Na procesoru ADSP21061 je k dispozici šest DMA kanálů - 4 přes sériové porty a 2 přes externí port. Programy mohou být nahrány na ADSP také pomocí DMA. Dále lze využít přerušení generované při dokončení DMA přenosu a DMA zřetězení (chaining) pro automatické spojení DMA přenosů.

- **Sériové porty (Serial Ports)**

Dva synchronní sériové porty poskytují rozhraní pro širokou škálu digitálních a smíšených periferních zařízení. Dokáží pracovat na frekvenci jádra, každý poskytující maximální datovou rychlost 40 Mbit/s. Nezávislé vysílací a přijímací funkce poskytují větší flexibilitu pro sériovou komunikaci. Data na sériovém portu mohou být automaticky přenášena z nebo do vnitřní paměti přes DMA. Díky časovému pipeline (time divide mode) umožňují sériově připojit více zařízení k jednomu portu.

- **Zavedení programu (Program Booting)**

Vnitřní paměť procesoru může být zavedena buď z 8-bitové EPROM nebo z řídicího počítače.

- **Přerušení (Interrupts)**

Protože DSP je určený pro operace v real-time systémech, je pro něj rozhodující účinná, propracovaná a předvídatelná přerušovací obsluha. ADSP21061 obsahuje 32 maskovatelných přerušení. Jejich zdroje mohou být vnitřní, vnější a programové. Každé přerušení má svoji pevnou prioritu. Procesor dovoluje i obsluhu vnořených přerušení (nesting mode). Může tak být přerušen i výkon obslužného programu předchozího přerušení. Jedinou podmínkou je, že nové přerušení musí mít vyšší prioritu (stejná nestačí). Kontrolu nad přerušením obstarává dvojice registrů – zámku (IRPTL) a masky (IMASK).

- **Řadič programu (Program Sequencer)**

Tento blok generuje tok adres instrukcí a umožňuje pružně kontrolovat běh programu. Zajišťuje provádění cyklů s nulovým režijním časem, podmíněné i nepodmíněné větvení (skoky) a poskytuje výkonný přerušovací systém.

- **Časovač (Timer)**

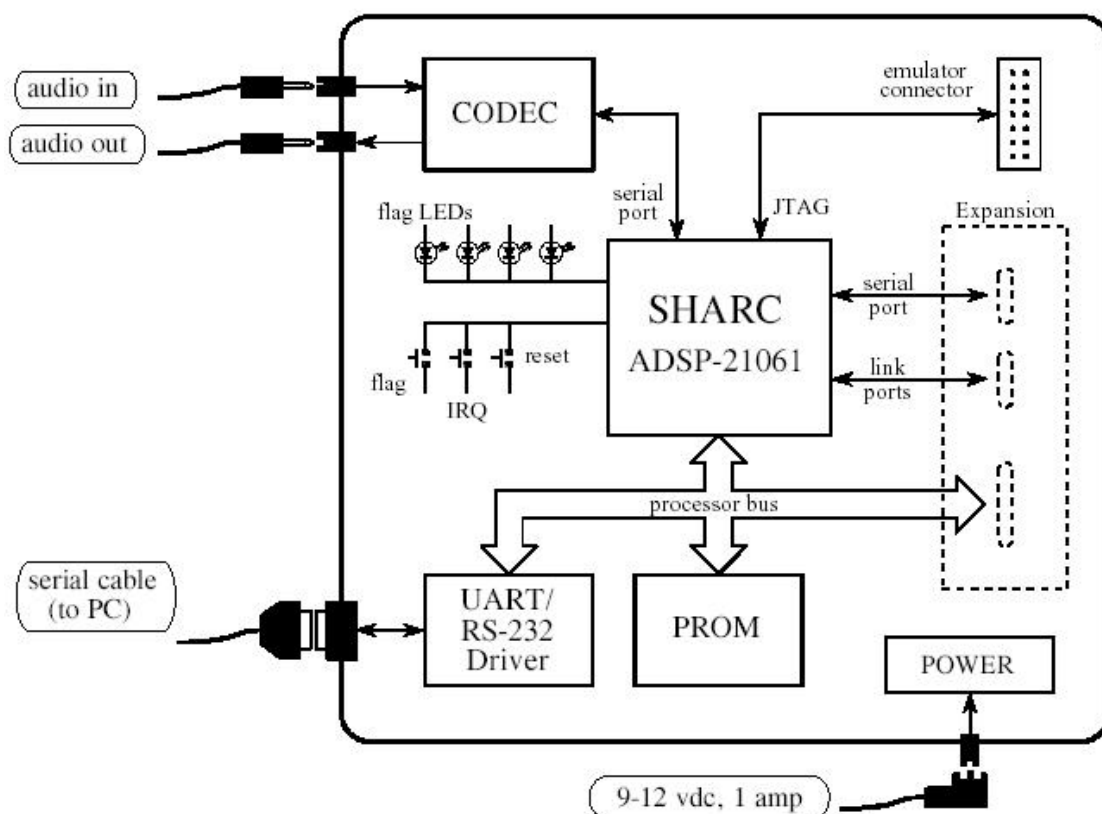
Časovač s programovatelným intervalem generuje periodické přerušení. Když je aktivovaný, dekrementuje v každém cyklu 32-bitový číselný registr. Dosáhne-li tento registr nuly, ADSP21061 vygeneruje přerušení [10,16,20,21].

5.2. Vývojová deska SHARC EZ-KIT Lite



obr.5.2 Vývojová deska SHARC EZ-KIT Lite

Procesor ADSP21061 se nachází na desce SHARC EZ-KIT Lite, která poskytuje potřebné rozhraní a nezbytná hardwarová rozšíření. Mimo procesoru, běžícím na 40 MHz, je součástí desky i převodník AnalogDevice1847, s kterým je deska přímo použitelná i pro praktická zpracování analogových signálů. Jedná se o 16-bitový sigma-delta převodník schopný digitalizovat dva vstupní kanály (stereo) rychlostí až 48 kHz a současně dva výstupní kanály na stejné rychlosti. Obvod AD1847 je pevně připojen k sériové lince SPORT0. Obvod UART chipu PC16550 zajišťující komunikaci po RS232 je připojen jako externí paměť přes adresovou a datovou sběrnici. Obě periferie jsou programovatelné, stejně tak parametry jejich interface. Blokové schéma vývojového kitu SHARC EZ-KIT Lite je na obr.5.3.



obr.5.3 Blokové schéma SHARC EZ-KIT Lite

Deska je dodávána i s potřebným softwarovým vybavením. Jde zejména o kombinovaný compiler assembler/C, linker, loader, monitor [10,22].

5.3. Programování

Kompilátor pro ADSP21061 umožňuje kompilovat zdrojový kód psaný programovacími jazyky: assembler, C a smíšeným jazykem assembler/C. Návrhy pro DSP bývají programovány v tomto pořadí: nejprve je napsán softwarový prototyp a je odladěn ve vysokoúrovňovém jazyce. Tento prototyp často poskytuje náležitý výkon. Jestliže je požadován větší výkon, tak je třeba najít ve vysokoúrovňovém kódu sekce potřebující nejvíce výpočetního času. Tyto kritické sekce jsou potom napsány v assembleru [19].

5.3.1. Využití programové prostředí

Samotný program wavelet analyzátoru byl programován v jazyce C s využitím standardních ANSI funkcí a specifických funkcí z knihovny ANALOG DEVICE. Pro zkompilování a nahrání programu do paměti procesoru byl využit software dodávaný s kitem ADSP.

Kompilátor G21K zkompiluje C zdrojový kód do assembleru, ten převede na soubor strojového kódu a poté linkuje tento strojový kód do jednotného souboru.

Obslužný program DIAG21K umožňuje základní řízení signálového procesoru a běhu jeho programu formou jednoduchých příkazů (zápis/čtení dat, nahrávání programu, reset, nastavení komunikace...) [10,23].

6. Realizace wavelet analyzátoru

6.1. Algoritmy DWT

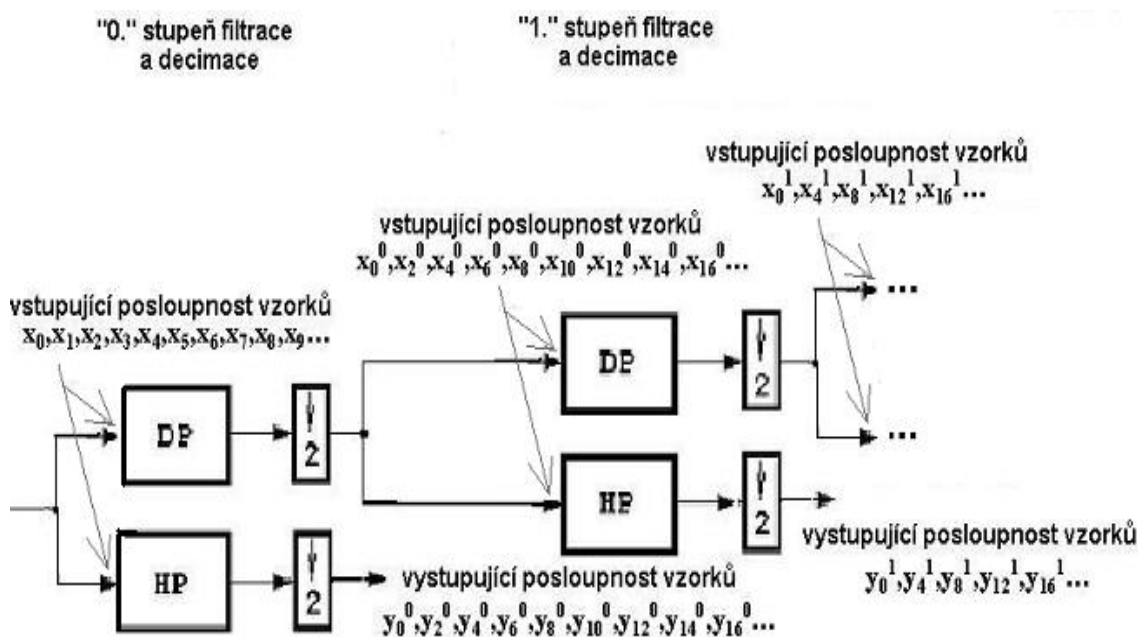
Přístupy ke konkrétnímu algoritmu výpočtu DWT podle obrázku 4.6 se dají rozdělit na dva zásadní. První oddělí rutinu vzorkování od vlastního zpracování – obdoba „post processing“, druhý realizuje všechny procesy, tedy i sběr vzorků, v jedné programové sekci – obdoba „real-time processing“.

Algoritmus řešící vlastní zpracování a vzorkování odděleně užívá dva procesy. Prvním je vzorkovací rutina, ve které se však řeší pouze ukládání vzorků. K paměti určené k této archivaci přistupuje v určitých intervalech proces druhý provádějící vlastní číslicové filtrace. V průběhu výpočtu se však nesmí data ztrácet. Standardně se tato situace řeší udáním priority jednotlivým procesům, tak aby rutina vzorkovače mohla podprogram výpočtu přerušit a vykonat uložení vzorku.

Druhý algoritmus řešící celý výpočet v rutině vzorkovače staví na faktu, že celkový počet vzorků zpracovávaných ve všech oktávách je roven součtu geometrické řady:

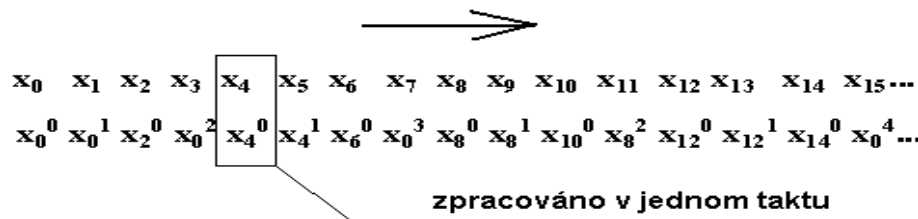
$$N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \dots = N \cdot \left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots\right) = N \cdot \frac{1}{1-0.5} = 2N \quad (6.1)$$

Tento vzorec vyplývá ze snižování počtu vzorků na polovinu při decimaci vždy po každé filtraci.



obr.6.1 Snižování počtu vzorků při DWT

Ze vztahu 6.1 je vidět, že celkový počet vzorků zpracovávaných ve všech oktávách kromě té nejvyšší (první) je roven N , což také odpovídá počtu vzorků zpracovávaných v samotné nejvyšší oktávě. To znamená, že pokud se zpracuje vzorek z A/D převodníku (tj. vzorek v nejvyšší oktávě) a jeden další vzorek v každé vzorkovací periodě, je možné vytvořit paralelní real-time spektrum pro všechny oktávy. Potom není potřeba ukládat data k následnému zpracování, ale je možné je analyzovat okamžitě, přímo v přerušovací rutině vzorkovače. První vzorek je představován vždy nejnovějším vzorkem $x_n = x[n]$, druhý x_n^d se bere z bufferu, kam byl uložen vzorek z předchozího stupně decimace atd. Horní index d říká, z kterého stupně decimace se vzorek bere [10,24].



obr.6.2 Pořadí při zpracování dat DWT

6.2. Algoritmus DWT pro ADSP21061

Pro výpočet DWT koeficientů byl zvolen algoritmus, který realizuje sběr vzorků a jejich zpracování v jedné programové sekci.

Postup je následující: Každý příchozí vzorek ze vzorkovače AD1847 vyvolá obslužnou funkci přerušování od sériového kanálu 0 a je uložen do proměnné *input*. Během přerušování je třeba zpracovat dva vzorky (viz. kapitola 6.1). Se vzorkem uloženým v proměnné *input* a s koeficienty wavelet filtru dolní a horní propusti je provedena konvoluce. Příslušné výsledky jsou uloženy do proměnných *dp_buf[0]* a *hp_buf[0]*. Tímto je analyzovaná frekvenční oblast rozdělena na dvě poloviny – *hp_buf[0]* představuje výstup nultého stupně rozkladu DWT analyzátoru a *dp_buf[0]* tvoří vstup pro zpracování prvního stupně.

Do proměnné *k* je uložena konstanta z pole *stage*. Toto pole obsahuje číslíčky udávající pořadí, pro který stupeň rozkladu se bude zpracovávat druhý vzorek (viz. horní indexy na obr.6.2). Jeho velikost *n_cycle* je odvozena od počtu zpracovávaných oktáv M ($n_cycle = 2^{M-1}$) a je do programu vloženo z hlavičkového souboru *UROVNE.H*. Tento soubor je generován samostatným programem psaným v C. Jeho zdrojový kód je na příloženém disku CD-R. Druhý zpracovaný vzorek je tedy uložen do proměnných

$dp_buf[k]$ a $hp_buf[k]$ a je výsledkem konvoluce wavelet filtrů dolní a horní propusti se vzorkem vystupujícím z dolní propusti o stupeň níže ($dp_buf[k-1]$).

```

/*obslužny program preruseni prijmu od Serioveho kanalu 0 */
void spr0_asserted( int sig_num ){
    int k;
    float input;

    /*nacteni aktualniho vzorku pro vypocet konvoluce na nultem stupni*/
    input = rx_buf[1];

    /*nacteni poradnice nasledujiciho vyssiho stupne*/
    k = stage[inc];

    /* nuly stupen */
    dp_buf[0] = fir(input, &coeffD8_lp[0], &stateD8_lpp[0][0], (int)N_COEFF);
    hp_buf[0] = fir(input, &coeffD8_hp[0], &stateD8_hpp[0][0], (int)N_COEFF);

    /* vyssi stupen */
    dp_buf[k] = fir(dp_buf[k-1], &coeffD8_lp[0], &stateD8_lpp[k][0], (int)N_COEFF);
    hp_buf[k] = fir(dp_buf[k-1], &coeffD8_hp[0], &stateD8_hpp[k][0], (int)N_COEFF);

    /* kruhovy buffer v poctu cyklu uzavirajicich vypocet N_STAGE oktavovych pasem */
    inc = (inc + 1) % N_CYCLE;
}

```

př.6.1 Výpočet DWT koeficientů pomocí fir filtrace

Výpis některých dalších důležitých funkcí programu a jejich popis je uveden v příloze a celý zdrojový kód programu *DWT.C* i s potřebnými hlavičkovými soubory je na přiloženém disku CD-R.

6.2.1. Funkce fir

Funkce *fir()* implementuje filtr s konečnou impulsní odezvou FIR definovaný svými koeficienty a zpožďovací linkou, které jsou dodány při volání této funkce. Struktura je následující:

float fir (float sample, float pm coeffs[], float dm state[], int taps).

Funkce vytváří filtrovanou odezvu vstupních dat. *Taps* udává délku filtru, což je současně délka pole koeficientů. Pole koeficientů *coeffs[]* je uloženo v programové paměti datového prostoru, aby byl umožněn dvojí přístup procesoru do paměti v jednom cyklu. Pole stavů *state[]* uložené v datové paměti obsahuje jako svůj první element ukazatel na zpožďovací linku následovaný hodnotami zpožďovací linky. Délka pole *state[]* je proto

o jednotku větší než délka filtru. Každý filtr musí mít své vlastní pole stavů, které by mělo být vynulováno před prvním voláním funkce *fir*.

```
float pm coeffD8_lp[MAX_COEFF] =
{
    #include "coeffD8_lp.h"
};

float pm coeffD8_hp[MAX_COEFF] =
{
    #include "coeffD8_hp.h"
};

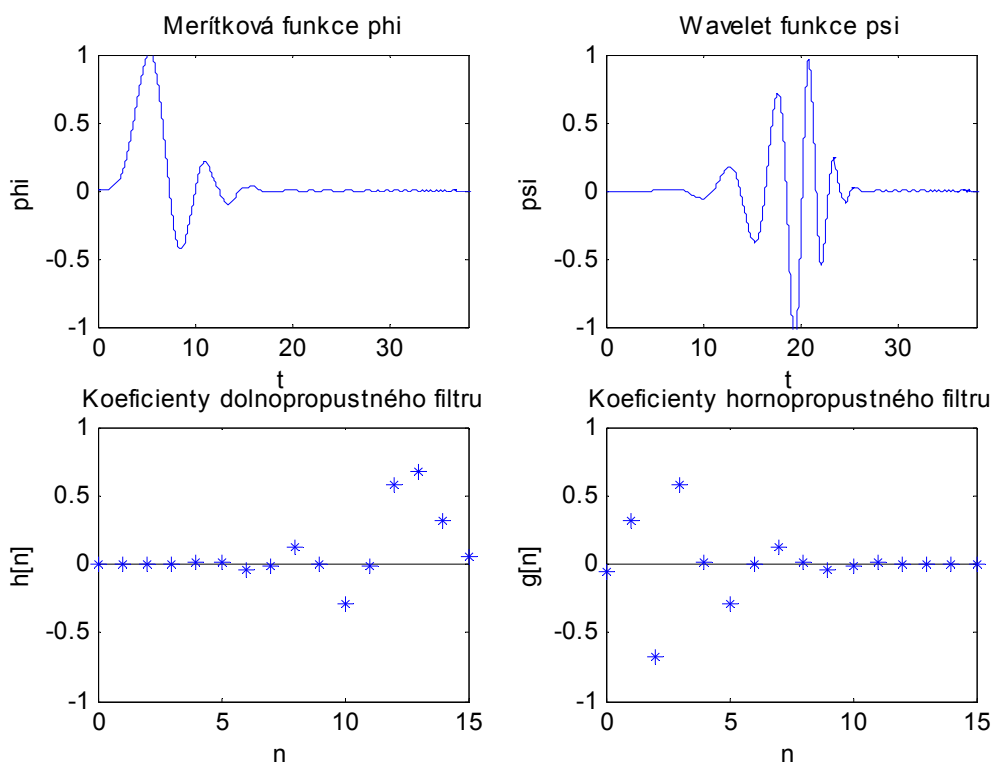
int stage[MAX_CYCLE]=
{
    #include "urovne.h"
};
/* +1 koeficient pro vnitřní potřebu funkce fir() */
float dm stateD8_lpp[N_STAGE][N_COEFF+1];
float dm stateD8_hpp[N_STAGE][N_COEFF+1];
```

př.6.2 Deklarace pole koeficientů do programové paměti *pm* a pole stavů do datové *dm*

Doba výpočtu FIR filtru je 25ns/tap. Funkce *fir()* je deklarována v hlavičkovém souboru *FILTERS.H*. Samotný kód pro výpočet funkce *fir()* je obsažen v assemblerovském modulu *FIR.ASM*. Ještě jsou potřeba dva další soubory: soubor *GENERIC.ACH* je soubor struktury, který definuje hardware ve smyslu paměťových prostorů a periférií, a soubor obsahující seznam koeficientů filtru [25].

6.2.2. Filtry DWT

V programu je pole koeficientů dolnoproustného filtru nazváno *coeffD8_lp* a je do programu vloženo z hlavičkového souboru *COEFFD8_LP.H* a pole koeficientů hornoproustného filtru nazváno *coeffD8_hp* je do programu vloženo z hlavičkového souboru *COEFFD8_HP.H*. Jedná se o koeficienty diskretních impulsních odezev filtrů odvozených od měřítkové funkce waveletu typu Daubechies osmého řádu. Hodnoty těchto koeficientů byly získány z wavelet toolboxu programu Matlab.

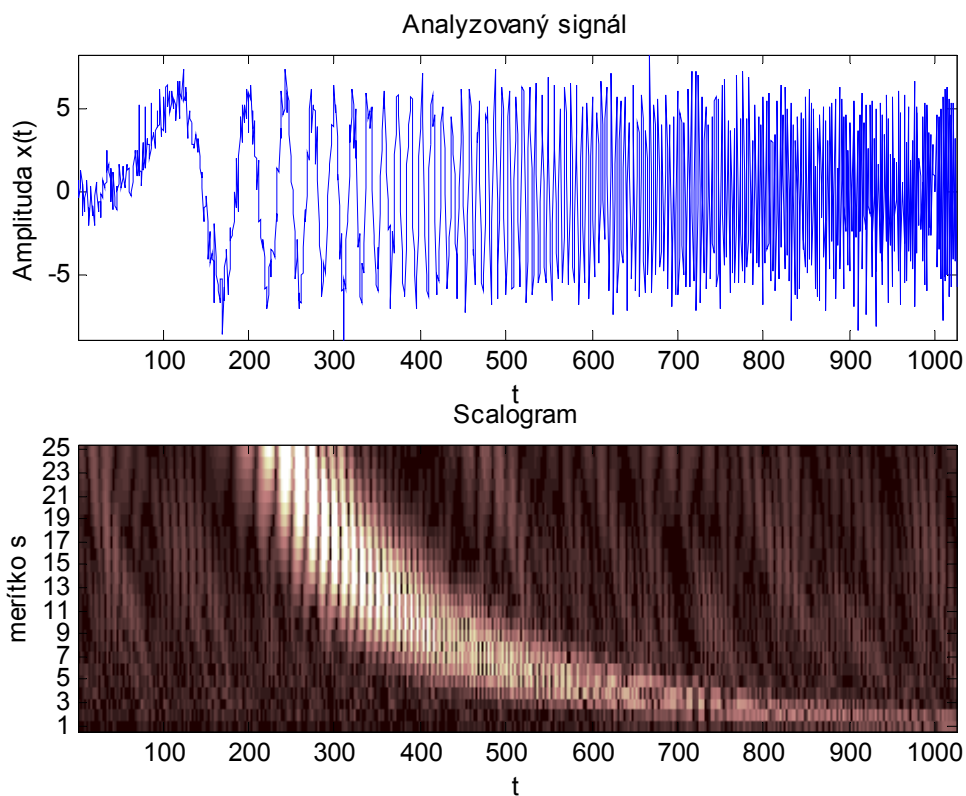


obr.6.3 Měřtková funkce a wavelet Daubechies 8, a koeficienty filtrů

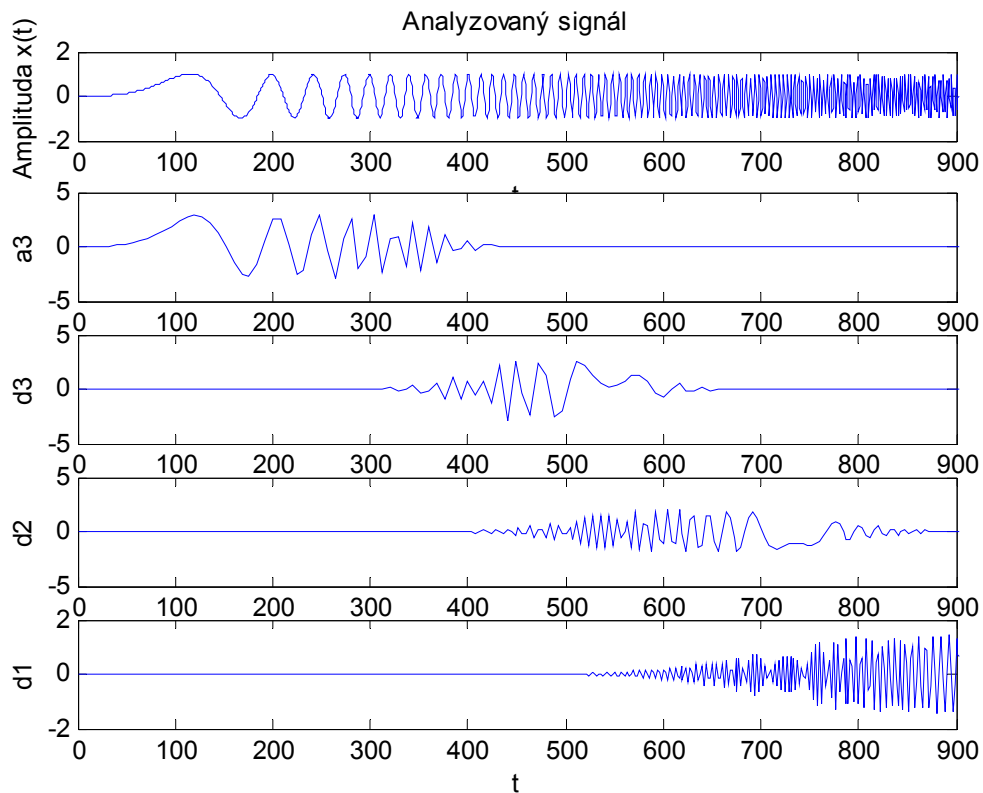
6.3. Možnosti grafického zobrazení

Je několik možností jak graficky zobrazit výsledky wavelet analýzy. Zobrazení pomocí tzv. scalogramu, získaného z CWT, lépe ukazuje jak se frekvence (měřítko s) mění s časem (viz. obr.6.4). Zobrazení průběhů aproximační a detailních složek, získaných DWT dekompozicí, lépe detekuje nespojitosti v signálu (viz. obr.6.5). Stupeň dekompozice můžeme určit podle toho, ve které aproximační složce se již neobjevuje šum.

Časovou proměnlivost frekvenčních spekter nestacionárního signálu lze také dobře graficky vyjádřit pomocí tzv. multispektra. Jde o uspořádání do 3D grafu s jednotlivými spektry za sebou tak, že vznikne další souřadnice vyjadřující časový okamžik, ve kterém je analýza provedena. Pro tyto grafy se užívá název spektrální mapy nebo zkráceně anglický termín waterfall (viz obr.7.2). Jiné uspořádání třetí souřadné osy může být představováno barevnou stupnicí nebo např. kroužky o poloměru úměrném velikosti složky (viz obr.3.2) [4].



obr.6.4 Zobrazení wavelet analýzy pomocí scalogramu



obr.6.5 Zobrazení aproximační a detailních složek signálu (3 stupně dekompozice)

7. Praktická měření

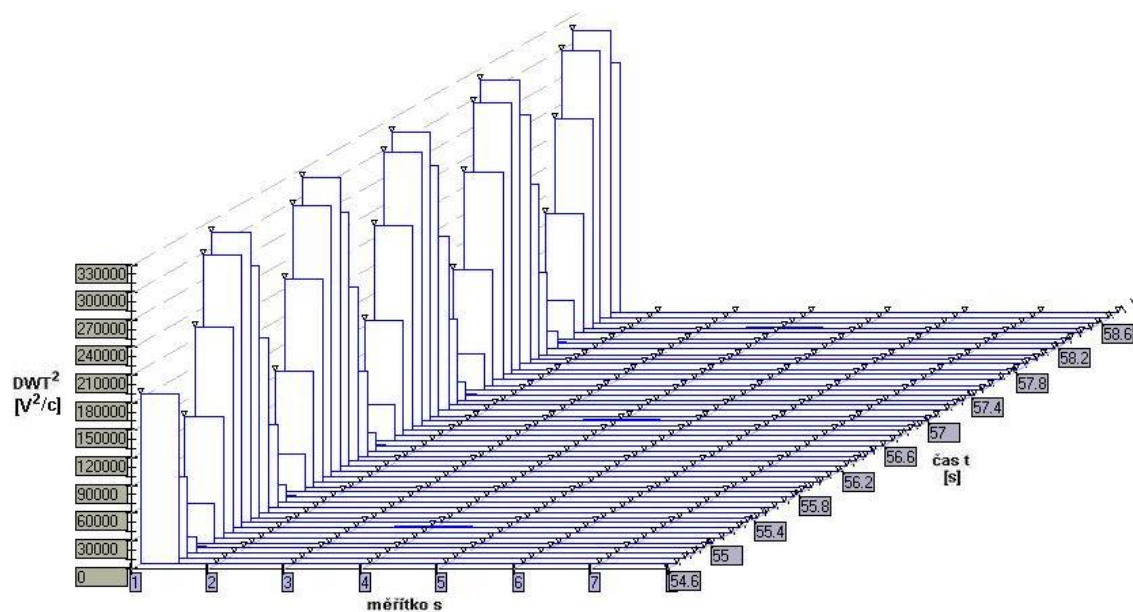
Samozřejmostí spektrální analýzy je její odpovídající vizualizace. Vlastní analyzátor, realizovaný na signálovém procesoru, je proto nutné doplnit zobrazovací jednotkou. Nejjednodušší je nechat signálový procesor připojený k hostitelskému počítači (PC) a na něm vytvořit program, který by zvládl sériovou komunikaci po RS232 s ADSP21061. K tomuto byl použit programový terminál poskytnutý Katedrou měření.

Signálový procesor zpracovává data v reálném čase. Výsledky analýzy jsou v *main* funkci formátovány z 32bitového typu *float* na typ *int* a jsou předávány do terminálu pomocí sériové linky. Pro lepší časové rozlišení vizualizace je možné výsledky před přenosem po RS232 formátovat na 8bitový typ *char*. Tím se sice sníží přesnost zobrazovaných hodnot, ale požadavek přenosu bude rychleji vyřízen. Ke komunikaci s ADSP21061 terminál využívá programové nadstavby kernelu ADSP. To je systém, který se automaticky nahraje do procesoru z paměti EPROM po připojení napájení a zajišťuje správnou inicializaci procesoru a sériovou komunikaci po RS232.

```
for(;;){  
  
    for(k=0; k<N_STAGE; k++){  
  
        ihp_buf[k] = hp_buf[k];           /*formatovani do int*/  
        chp_buf[k] = (ihp_buf[k]>>8) & 0x00ff; /*formatovani do char*/  
    }  
}
```

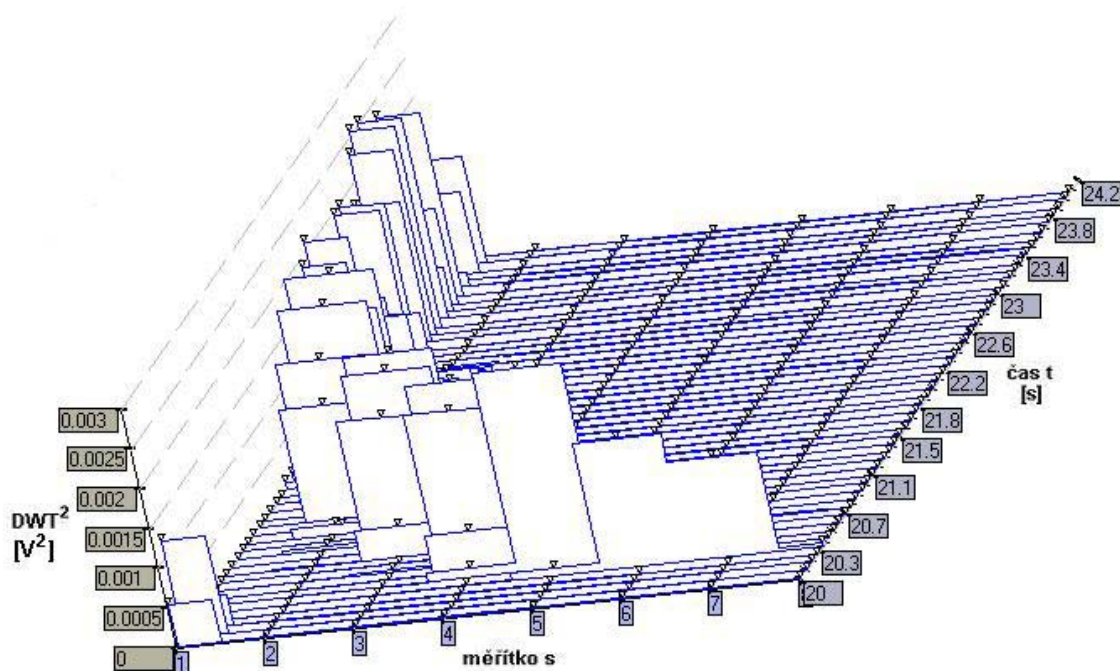
př.7.1 Formátování výsledků DWT na typ *char* pro rychlejší přenos po RS232

DWT koeficienty získané z ADSP21061 z pole *ihp_buf[k]* jsou v programu terminálu umocněny na druhou a vynásobeny kalibrační konstantou *c*, aby vyjadřovaly skutečný okamžitý výkon. Terminál graficky zobrazuje časový průběh okamžitého výkonu koeficientů DWT jednotlivých úrovní dekompozice. Vzniká tak třírozměrný graf typu waterfall závislostí výkonu waveletových koeficientů, měnících se v čase, na měřítku *s*.

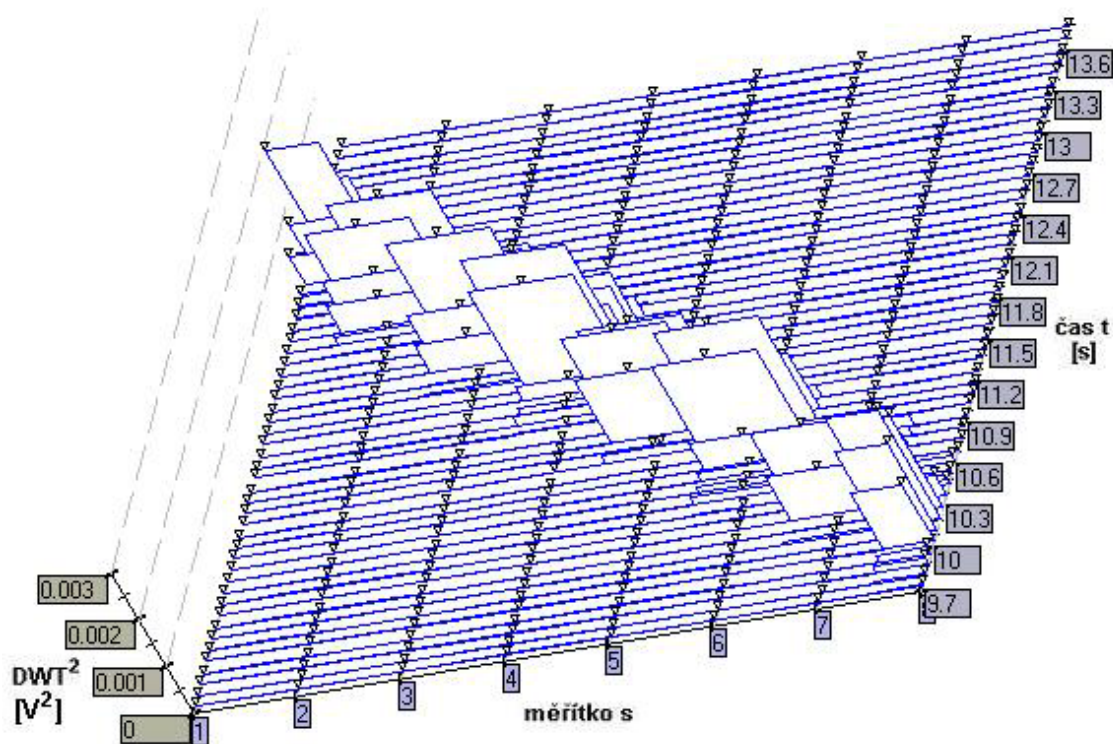


obr.7.1 Necejchovaná wavelet analýza harmonického signálu 6kHz

Před testováním funkčnosti bylo nutné provést elektrické cejchování wavelet analyzátoru a zvolit kalibrační konstantu c . Signály pro cejchování i modelové signály pocházely z funkčního generátoru HP33120A, byly přivedeny na vstup *line2* převodníku AD1847, jehož vstupní piny musely být pro tento účel připájeny k desce EZ-KIT. Výsledky wavelet analýzy modelových nestacionárních signálů jsou graficky představeny na obrázku 7.2 a 7.3.



obr.7.2 DWT lineárně frekvenčně přelad'ovaného sinusového signálu

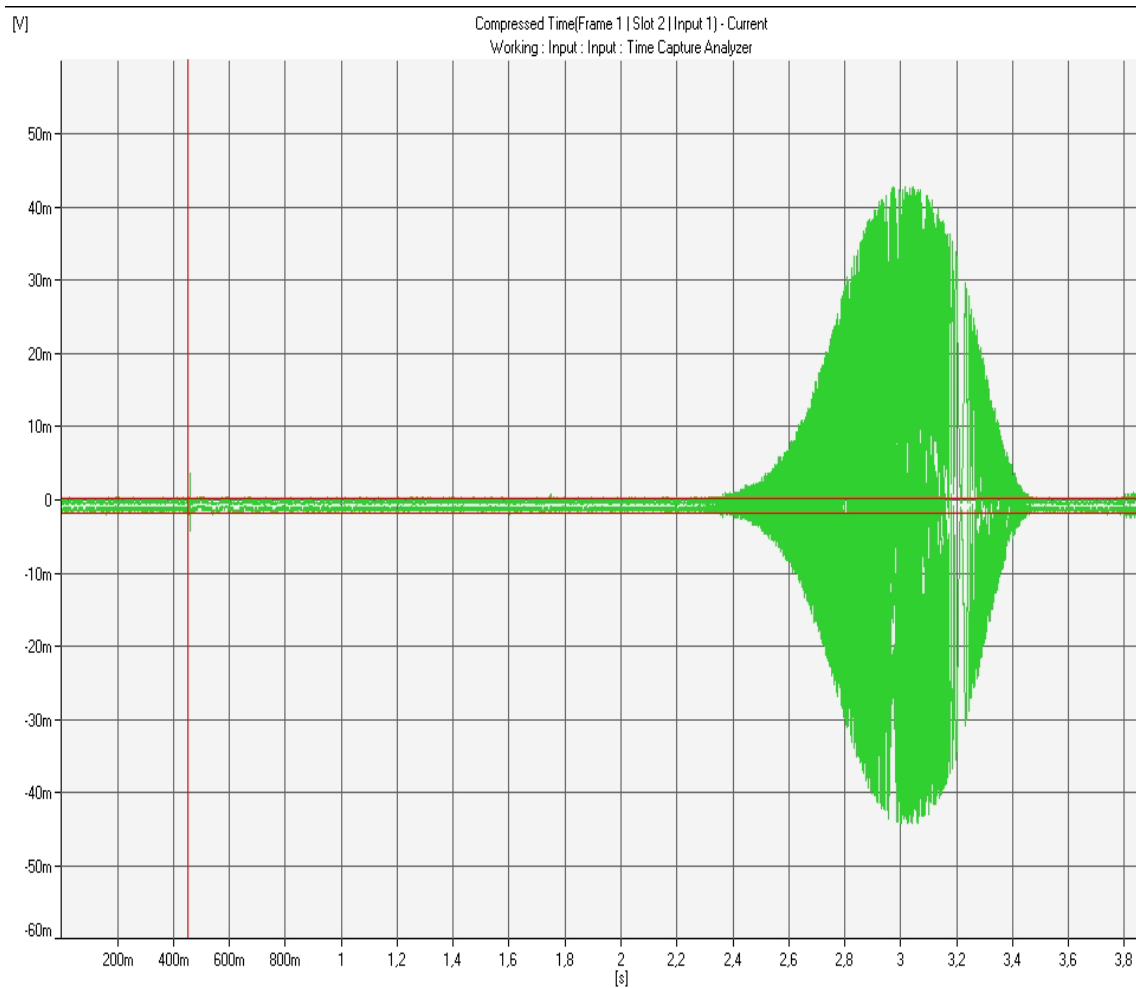


obr.7.3 DWT logaritmicky frekvenčně přeladovaného sinusového signálu

Na obr.7.2 a obr.7.3 jsou krátké časové výseky z real-time wavelet analýzy sinusových signálů, jejichž frekvence vzrůstá v čase.

Problémem při zobrazování reálného průběhu DWT koeficientů se ukázala nedostačující rychlost přenosu dat po RS232 z desky EZ-KIT do PC, na kterém běží vizualizační terminál. I při využití maximální přenosové rychlosti linky RS232 (115 200b/s) a snížení velikosti přenášeného slova na 8 bitů a vzorkovací frekvence na minimum se na terminálu nezobrazí plnohodnotná DWT, neboť dochází ke ztrátě dat způsobené právě nedostačující rychlostí přenosu. Tato skutečnost omezuje použití tohoto wavelet analyzátoru pouze pro pomalé signály a analyzátor se proto jeví nepoužitelný pro zjištění krátce trvajících nestacionarit.

Na obr.7.4 je časový průběh DWT koeficientů z první úrovně dekompozice (tj. první detailní složka signálu) získaný DWT analyzátozem analýzou lineárně frekvenčně přeladovaného signálu. Pro zobrazení byla použita jednotka PULSE frontend, k níž byla deska EZ-KIT připojena pomocí audio výstupu (audio out).



obr.7.4 Časový průběh DWT koeficientů z první úrovně dekompozice

8. Závěr a zhodnocení

Práce splnila všechny body, tak jak jsou popsány v zásadách pro vypracování. Teoreticky byly popsány vlastnosti a metody frekvenční transformace a časově frekvenční transformace, jejich výhody, uplatnění a vzájemné srovnání. Zvláště jsou popsány základní principy a vlastnosti waveletové transformace.

Cílem bylo naprogramovat na signálovém procesoru ADSP21061 algoritmus výpočtu diskrétní waveletové transformace. Program wavelet analyzátoru využívající číslicové filtrace byl navržen pro zpracování signálu v reálném čase a využívá algoritmus, který realizuje sběr vzorků a jejich zpracování v jedné programové sekci. Funkčnost analyzátoru byla vyzkoušena na modelových nestacionárních signálech. Použitelnost analyzátoru využívající grafickou vizualizaci na PC je bohužel limitovaná pouze pro analýzu pomalých signálů v důsledku nedostačující rychlosti přenosu dat po RS232 z desky EZ-KIT do PC, na kterém běží vizualizační terminál.

Grafického představení plnohodnotné DWT se všemi koeficienty by se dalo dosáhnout rychlejším připojením desky EZ-KIT k vizualizační jednotce. Například s využitím vyvedené programové a datové sběrnice připojit k desce rozhraní pro paralelní přenos dat.

Literatura:

- [1] Vích, R. – Smékal, Z.: Číslíkové filtry, Academia Praha, 2000
- [2] Fiala, J.: Waveletová transformace,
<http://www.xray.cz/ms/bul99-1/fiala.pdf>
- [3] Eisenstová, G. – Sedláček, M.: Využití Matlabu k potlačování aditivního šumu pomocí filtrace a pomocí vlnkové transformace,
<http://phobos.vscht.cz/MATLAB01/eisenst.pdf>
- [4] Tůma, J.: Zpracování signálů získaných z mechanických systémů užitím FFT, Sdělovací technika Praha, 1997
- [5] Fourierova transformace, Učební texty Univerzity Palackého v Olomouci,
<http://apfyz.upol.cz/ucebnice/down/mini/fourtrans.pdf>
- [6] Nouza, J.: Přednášky z předmětu Zpracování, přenos a identifikace signálů na TUL, FM 4. ročník, 8. semestr,
http://www.fm.vslib.cz/~kes/pages/sig/ramce_main.html
- [7] Šumbera, J.: Aplikace vlnkové transformace v texturní faktografii únavových lomů, Diplomová práce, FJFI ČVUT Praha, 2003,
<http://www.jikos.cz/%7esumbera/vyplody/diplomka.pdf>
- [8] Blaška, J.: Moderní metody analýzy signálů,
<http://www.sh.cvut.cz/~blaska/>
- [9] Jaksch, I.: Přednášky z předmětu Číslíkové měřicí systémy na TUL, FM 4. ročník, 8. semestr,
http://www.fm.vslib.cz/~kamm/cms/10_zakl_freq_anal.pdf
- [10] Mareš, J.: Návrh CPB analyzátoru vibrací pomocí signálového procesoru, Diplomová práce, TUL Liberec, 2003
- [11] Smutný, J.: Časově frekvenční analýza stavebních konstrukcí a materiálů, Zkrácená verze habilitační práce, VUT Brno, 2002,
<http://www.vutbr.cz/nakl/habilit/Smutnyw.pdf>
- [12] Shie Qian - Dapang Chen: Joint Time-Frequency Analysis: Method and Application, Prentice Hall PTR, 1996
- [13] Semotán, J.: Demonstrace transformací, Diplomová práce, ČVUT Praha, 2001
- [14] Maršálek, T. – Merta, M.: Vlnková analýza,
<http://home.zcu.cz/~tmarsal/kro/wavelet/wavelet.html>

- [15] Šmíd, R.: Úvod do vlnkové transformace,
<http://measure.feld.cvut.cz/usr/staff/smid/wavelets/Wavelet-intro8859.pdf>
- [16] Sedláček, M.: Zpracování signálů v měřicí technice, ČVUT Praha, 1998
- [17] Kolář, R.: Předzpracování medicínských ultrazvukových tomografií, Zkrácená verze habilitační práce, VUT Brno, 2002,
<http://www.vutbr.cz/nakl/thesis/KolarRadw.pdf>
- [18] Provazník, I.: Vlnková analýza pro detekci signálů – aplikace v experimentálním kardiologickém výzkumu, Zkrácená verze habilitační práce, VUT Brno, 2002
<http://www.vutbr.cz/nakl/habilit/Provaznikw.pdf>
- [19] For efficient signal processing in embedded systems, take a DSP, not a RISC, Analog dialogue 30-3, 1996,
http://www.analog.com/library/analogDialogue/archives/30-3/Dialogue30_3pg3.pdf
- [20] ADSP-21061 SHARC Preliminary Data Sheet, Analog Devices, 1996
- [21] ADSP-2106x SHARC User's Manual, Analog Devices, 1996
- [22] Getting started with DSPs,
<http://bbs.chinaecnet.com/pdf/dsp01.pdf>
- [23] ADSP-2106x SHARC EZ-KIT Lite Reference Manual, Analog Devices, 1997
- [24] Randall, R.B.: Frequency Analysis, Brüel & Kjar, Narum, 1987
- [25] ADSP-21000 Family C Runtime Library Manual, Analog Devices, 1995
- [26] Analog Devices, Serial-Port 16-Bit SoundPort StereoCodec AD1847, 1996
<http://www.alsa-project.org/alsa/ftp/datasheets/ad/ad1847.pdf>

Příloha 1:**Výpis důležitých funkcí programu DWT pro ADSP21061**

Vedle funkce vykonávající samotnou číslicovou wavelet transformaci, bylo nutné naprogramovat nastavení periferních zařízení a interních přenosových cest DMA a sériového portu.

DMA řadič 0 je určen k obsluze příjmu od sériového portu SPORT0 (buffer RX0), DMA řadič 2 je určen k obsluze vysílání (buffer TX0). Na SPORT0 je připojen A/D převodník AD1847 s kompatibilním sériovým rozhraním. Sériové porty ADSP21061 jsou synchronní, mohou pracovat až na frekvenci jádra. Jejich nastavení je široce variabilní (softwarově i hardwarově), aby tak bylo možné připojit co nejširší okruh přístrojů. Nastavitelné parametry jsou sdruženy ve 4 základních registrech oddělených pro příjem a pro vysílání:

Řídící registry vysílání a příjmu	SRCTLx, STCTLx
Registry časování a synchronizace	RDIVx, TDIVx
Registry časového multiplexu	MRCSx, MTCSx
Registry pro nastavení kvantizace bitů	MRCCSx, MTCCSx

tab.p1 Tabulka řídicích registrů SPORTx

Jejich nastavení pro potřeby programu DWT analýzy je následující:

```
void setup_sports ( void )
{

    /*nastavení vícekanálové - multichannel komunikace */
    sport0_iop.mtcs = 0x00070007;    /* vysílání dat na pozicích 0, 1, 2, 16, 17, 18
                                     sériového slova */
    sport0_iop.mrcs = 0x00070007;    /*příjem dat na pozicích 0, 1, 2, 16, 17, 18
                                     sériového slova */
    sport0_iop.mtccs = 0x00000000;    /* vysílání bez komprese */
    sport0_iop.mrccs = 0x00000000;    /* přijímání bez komprese */

    /* ŘÍDÍCÍ REGISTR VYSÍLÁNÍ */
    /* STCTL0 <= 0x001c00f */

    sport0_iop.txc.mdf = 1;    /* zpoždění mezi synchronizačním bitem a prvním
                               datovým bitem (MFD) */
    sport0_iop.txc.schen = 1;    /* Tx DMA zřetězení povoleno */
    sport0_iop.txc.sden = 1;    /* Tx DMA povoleno */
}
```

```

sport0_iop.txc.lafs = 0; /* zpoždění TFS = transfer frame sync */
sport0_iop.txc.ltfs = 0; /* aktivuje nízkou logickou úroveň TFS */
sport0_iop.txc.ditfs = 0; /* nezávislost dat TFS */
sport0_iop.txc.itfs = 0; /* určuje zdroj TFS */
sport0_iop.txc.tfsr = 0; /* určuje zda je TFS vyžadováno */

sport0_iop.txc.ckre = 0; /* data a FS reagují na vzestupnou hranu hodin */
sport0_iop.txc.gclk = 0; /* povolení hodin pouze v době přenosu */
sport0_iop.txc.iclk = 0; /* vnitřní generování hodinového signálu pro vysílání */
sport0_iop.txc.pack = 0; /* rozložit 32b slovo do dvou 16b */

sport0_iop.txc.slen = 15; /* délka dat - 1 */
sport0_iop.txc.sendn = 0; /* datový endian 1 = LSB první */
sport0_iop.txc.dtype = SPORT_DTYPE_RIGHT_JUSTIFY_SIGN_EXTEND;
/* specifikátor datového typu */
sport0_iop.txc.spen = 0; /* povolení */

/* ŘÍDÍCÍ REGISTR PŘÍJMU */
/* SRCTL0 <= 0x1f8c20f2 */

sport0_iop.rxc.nch = 31; /* počet kanálů - 1 pro multichannel */
sport0_iop.rxc.mce = 1; /* multichannel povoleno */
sport0_iop.rxc.spl = 0; /* zkratovací smyčka portů pro interní testování */
sport0_iop.rxc.d2dma = 0; /* povolení 2-rozměrné DMA - nelze u ADSP21061 */
sport0_iop.rxc.schen = 1; /* Rx DMA zřetězení povoleno */
sport0_iop.rxc.sden = 1; /* Rx DMA povoleno */
sport0_iop.rxc.lafs = 0; /* zpoždění RFS = transfer frame sync */
sport0_iop.rxc.ltfs = 0; /* aktivuje nízkou logickou úroveň RFS */
sport0_iop.rxc.irfs = 0; /* určuje zdroj RFS */
sport0_iop.rxc.rfsr = 1; /* určuje zda je RFS vyžadováno */
sport0_iop.rxc.ckre = 0; /* data a FS reagují na vzestupnou hranu hodin */
sport0_iop.rxc.gclk = 0; /* povolení hodin pouze v době přenosu */
sport0_iop.rxc.iclk = 0; /* vnitřní generování hodinového signálu pro příjem */
sport0_iop.rxc.pack = 0; /* složení dvou 16b přijatých slov do 32b slova */

sport0_iop.rxc.slen = 15; /* délka dat - 1 */
sport0_iop.rxc.sendn = 0; /* datový endian 1 = LSB první */
sport0_iop.rxc.dtype = SPORT_DTYPE_RIGHT_JUSTIFY_SIGN_EXTEND;
/* specifikátor datového typu */
sport0_iop.rxc.spen = 0; /* povolení */

/* Povolení přerušení sport0 (DMA povoleno) */
interrupt(SIG_SPR0I, spr0_asserted);
interrupt(SIG_SPT0I, spt0_asserted);

/* Nastavení vysílacího řídicího bloku pro zřetězené DMA */
tx_tcb.ii = tx_buf; /* zdroj vysílaných dat DMA kanálu */
tx_tcb.cp = &tx_tcb.ii; /*ukazatel následující DMA sekvence nastaven na sebe */
SetIOP(CP2, (((int)&tx_tcb.ii) & CP_MAF) | CP_PCI);
/* definuje pointr na současný TCB */

```

```
/* Nastavení přijímacího bloku pro zřetěžené DMA */
rx_tcb.ii = rx_buf;      /* zdroj přijímaných dat DMA kanálu */
rx_tcb.cp = &rx_tcb.ii;  /* ukazatel následující DMA sekvence nastaven na sebe */
SetIOP(CP0, (((int)&rx_tcb.ii) & CP_MAF) | CP_PCI);
/* definuje pointer na současný TCB */
}
```

Kromě nastavení řídicích registrů jsou v podprogramu také přiřazeny přerušení výkonným podprogramům pro příjem a zápis sériových dat. Dva poslední odstavce provádí nastavení některých parametrů TCB struktury, která řídí DMA přenos.

```
/* Řídicí blok pro zřetěžený DMA přenos-TBC(Transfer Control Block)*/
typedef struct {
    unsigned lpath3;      /* pouze pro zajištění multiprocessingu */
    unsigned lpath2;      /* pouze pro zajištění multiprocessingu */
    unsigned lpath1;      /* pouze pro zajištění multiprocessingu */
    unsigned db;          /* využíváný k 2D DMA přenosu */
    unsigned gp;          /* při zřetěžení využíváný jako ukazatel
                           na poslední DMA sekvenci */
    unsigned** cp;        /* ukazatel na následující DMA sekvenci */
    unsigned c;           /* čítač, říká kolik zbývá přenést */
    int im;               /* modifikátor, přičten k ii určuje pozici dalších dat */
    unsigned * ii;        /* pointer na data k přenosu */
} _tcb;

_tcb rx_tcb = {0, 0, 0, 0, 0, 0, 3, 1, 0}; /* přijímací TCB */
_tcb tx_tcb = {0, 0, 0, 0, 0, 0, 3, 1, 0}; /* vysílací TCB */
```

Tato struktura definuje zejména oblast paměti, se kterou má DMA řadič pracovat. Nejdůležitější je pointer aktuální pozice *ii*, který je v našem případě nutné na úvodu nastavit na adresu proměnné pro příjem/vysílání dat. Další zásadní položka *cp* určuje ukazatel na následující TCB strukturu, která se aplikuje po skončení tohoto DMA přenosu. Automatická návaznost DMA sekvencí je zajištěna mechanismem zřetěžení (chaining mode).

Celý mechanismus pak pracuje takto. Zaplnění přijímacího bufferu generuje DMA sekvenci, díky níž se slovo z bufferu запиše na místo v interní paměti definované v TCB. Uživatel tak nemusí obsluhovat čtení či zápis IO registrů RX a TX, ale může zpracovat stejná data v interní paměti. Po přenesení třech slov, na něž je nastaven count registr *c* TCB (vzorek z levého kanálu, pravého kanálu a control word), je sekvence skončená a generuje se přerušení o příjmu. Následuje další sekvence DMA, která je inicializovaná stejnou TCB strukturou (jde o příjem stejného typu dat – nových vzorků).

Generované přerušení je vhodné využít ke zpracování přijatých dat, neboť ty budou další sekvencí přepsány.

Procesor ADSP21061 je součástí vývojové desky SHARC EZ-KIT LITE. Ta je osazena obvodem UART16550 pro komunikaci po RS232, EPROM paměti, ze které se po přivedení napájení k desce zavádí jednoduchý systém zajišťující správnou inicializaci procesoru a sériovou komunikaci po RS232. Také se na ní nachází jednoduchý program demonstrující funkčnost.

Obvod UART je klasickým prvkem se známými parametry nastavení (druh řízení, přenosová rychlost, parita, stop bity...). Obvod celkově obsahuje 12 registrů – 4 řídicí, 4 stavové, 2 pro příjem a vysílání, 2 pro nastavení časovače. V případě že využijeme funkcí komunikačního interface kernelu, nemusíme se detailnějším studiem jejich nastavení zabývat.

Důležitým prvkem je také obvod vzorkovače AD1847. Jde o sofistikovaný obvod, variabilní v mnoha parametrech. Komunikace s ním se děje třemi slovy. Jako první se vysílá slovo řídicí, další jdou data pro levý a pravý kanál. Stejně je to i s přijímáním, první jde informace o stavu, další odchází vzorkovaná data z obou kanálů. Řídicím slovem také zapisujeme do 16 registrů provozní stav, jde o funkce uvedené v tabulce tab.p2.

Konkrétní nastavení vyhovující potřebám DWT analýzy je následující:

```
/* 0xc# značí zápis, do řídicího reg. s pořadovým číslem # */
/* 0x8# značí ukončení zápisu */
#define SZ_regs_1847 16
int regs_1847[SZ_regs_1847] = {
    0xc080, /* LeftInputControl (vstup z levého kanálu line2 bez zesílení) */
    0xc180, /* RightInputControl (vstup z pravého kanálu line2 bez zesílení) */
    0xc280, /* LeftAux#1InputControl (němý) */
    0xc380, /* RightAux#1InputControl (němý) */
    0xc480, /* LeftAux#2InputControl (němý) */
    0xc580, /* RightAux#2InputControl (němý) */
    0xc600, /* LeftDACControl (vystup je aktivní a není nijak tlumen) */
    0xc700, /* RightDACControl (vystup je aktivní a není nijak tlumen) */
    0xc852, /* DataFormat (lineární kvantování, stereo, XTAL1 24.5MHz s
                dělicím faktorem 1536 -> fvz = 16kHz) */
    0xc909, /* InterfaceConfigurationRegister (po skončení nastavení
                zapnuta autokalibrace) */
    0xca00, /* PinControl (TTL logic LO) */
    0xcb00, /* nevyužit */
    0xcc40, /* MiscellaneousInformation (Vysílání na pozicích 0,1,2 s délkou
                slova 32b ); */
    0xcd00, /* DigitalMixControl (neaktivován) */
```

0xce00, /* nevyužit */
 0x8f00}; /* nevyužit, v řídicím slově jsme ukončili nastavování*/

Index	Registr	Funkce
0	Levý vstup	Vstupní zesílení (0, 22.5)dB, volba levého vstupu
1	Pravý vstup	Vstupní zesílení (0, 22.5)dB, volba pravého vstupu
2	1. levý přídavný vstup	Zisk (+12, -34.5)dB, povolení/zákaz kanálu
3	1. pravý přídavný vstup	Zisk (+12, -34.5)dB, povolení/zákaz kanálu
4	2. levý přídavný vstup	Zisk (+12, -34.5)dB, povolení/zákaz kanálu
5	2. pravý přídavný vstup	Zisk (+12, -34.5)dB, povolení/zákaz kanálu
6	Levý D/A	Útlum (0, -94.5)dB, povolení/zákaz kanálu
7	Pravý D/A	Útlum (0, -94.5)dB, povolení/zákaz kanálu
8	Data formát	Zdroj hodin, dělicí faktor, stereo/mono, komprese, 8/16 bitů
9	Konfigurace interface	Playback, autokalibrace
10	Pin control	Tří stavový hodinový signál, citlivost na externí řídicí vstup
11	Invalid adress	Nevyužito
12	Různé	1 nebo 2 vodičový systém, délka sériového slova
13	Digitální mix	Nastavení útlumu (0 – 94.5)dB, jeho povolení/zákaz
14	Invalid adress	Nevyužito
15	Invalid adress	Nevyužito

tab.p2 Tabulka registrů AD1847
 [10,21,26]